Using deep learning to predict continuous hand kinematics from magnetoencephalographic (MEG) measurements of electromagnetic brain activity.

Matteo Anelli

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 31.12.2020

Supervisor

Prof. Lauri Parkkonen

Advisor

MSc. Ivan Zubarev

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Matteo Anelli

Title Using deep learning to predict continuous hand kinematics from magnetoencephalographic (MEG) measurements of electromagnetic brain activity.

Degree programme ICT Innovation - EIT Digital Master School

Major Data Science **Code of major** SCI3095

Supervisor Prof. Lauri Parkkonen

Advisor MSc. Ivan Zubarev

Date 31.12.2020 **Number of pages** 74 **Language** English

Abstract

Brain imaging techniques aim to study and discover hidden patterns from the brain activity that can lead to a better understanding of brain dynamics as well as to better clinical diagnoses. However, the brain is a complex system. Therefore, the encoded information is tedious to extract and analyze. In recent years, deep learning has outperformed state-of-the-art statistical techniques in different fields, such as computer vision and speech recognition. The reason for this is mainly its capability to extract complex patterns throughout an automatic end-to-end learning process. Thus, the main goal of the thesis was to investigate the potential and limitations of deep learning (DL) techniques to decode continuous hand-kinematics parameters from electromagnetic brain activity measured with magnetoencephalography (MEG).

The primary thesis experiment consisted of decoding circular hand-movement captured using accelerometers placed on the back of both subject hands. Specifically, the principal analysis was a within-subject experiment. The primary baseline approach used as a comparison to the DL proposed solutions is a state-of-the-art algorithm called Source Power Comodulation (SPoC), previously used to model regression tasks. It performs spatial filtering of the MEG data estimating the source space that maximally correlates with the continuous target value, and, eventually, uses the estimated source space to predict the target.

The two proposed models are Convolutional Neural Network (CNN) architectures that aim to extract meaningful features from the measurement by applying specific

transformations to the input data. The first proposed model (MNet) aims firstly to extract global features convolving simultaneously in the spatial and temporal domain of the recording. Secondly, it aims to extract local features. The second one is a Spatial CNN (SCNN) that aims to separately extract temporal and spatial features and, eventually, combine them to predict the final output. Moreover, to augment the input data to boost the performances, the Relative Power Spectrum (RPS) of some specific bands was integrated into the data input.

The main results have shown that the proposed models outperformed the SPoC algorithm in a within-subject experiment. Specifically, the final performances were the following: the SPoC had an RMSE of 0.976, the RPS-SCNN got an RMSE of 0.841, and the RPS-MNet got an RMSE of 0.428.

As a result, DL techniques specifically designed to work with MEG data outperformed the SPoC algorithm in decoding the continuous target variable. Consequently, DL-based application can provide a valuable alternative to decode hand-movement parameters from MEG measurements in a within-subject experiment.

Keywords Magnetoencephalography, Convolutional Neural Network, Source Power Comodulation, MEG, CNN, SPoC, motor encoding, sensorimotor rhythm

Acknowledgements

I would like to give thanks to all people that provided me with support during the research and writing of my thesis. Firstly, I would like to thank my advisor Ivan Zubarev for his tips and expertise in the area of neuroscience and machine learning. I would like to thank my supervisor professor Lauri Parkkonen for giving me the opportunity to do my thesis in his group and for his continuous guidance. I would also like to thank the Aalto Triton support team for the continuous support and guidance provided along the way. Finally, I would like to thank Hanna-Leena Halme for letting me use the principal dataset.

Otaniemi, 31.12.2020

Matteo Anelli

Author's Contribution

The author designed, implemented, and performed computational experiments with published deep-learning methods on existing neuroimaging datasets. The author was not involved in designing the experiment and data collection.

The code used to perform the thesis experiments is available at: https://github.com/matteoanelli/MEG-prediction.

Contents

Abbreviations

AE	Autoencoder
ANN	Artificial Neural Network
BCI	Brain–computer interface
CNN	Convolutional Neural Network
CNS	Central Nervous System
CT	Computerized Tomography
ECoG	Electrocorticography
EEG	Electroencephalography
EP	Evoked Potential
FCFFNN	Fully Connected Feedforward Neural Network
fMRI	Functional Magnetic Resonance Imaging
ICA	Independent Components Analysis
M1	Primary Motor Cortex
MEG	Magnetoencephalography
MRI	Magnetic Resonance Imaging
NN	Neural Network
PNS	Peripheral Nervous System
PSP	Postsynaptic Potentials
RNN	Recurrent Neural Network
RMSE	Root Mean Square Error
SMA	Supplementary Motor Area
SNR	Signal-to-Noise Ratio
SQUID	Superconducting Quantum Interference Devices

1 Introduction

Neurons work together in complex networks to generate phenomena, such as thoughts and feelings. They exchange information using chemical and electrical signals. The electrical signals carried along neurons are called action potentials. Even though the action potential is generated inside each of them, neurons are organized in neural circuits of different dimensions and with different functionality. As a result, multiple neurons fire together, generating a chain of multiple simultaneous action potentials. These repeated cycles of organized electrical activity are called neural oscillations.

Neuroimaging techniques use specific neural oscillations to examine the underlying neural activity. Various brain imaging techniques differ in characteristics, such as signal sources, energy levels, spatial resolution, and temporal resolution; each technique has particular advantages and disadvantages. Hence, different techniques provide complementary information. The two datasets studied during this thesis are composed of Magnetoencephalography (MEG) and Electroencephalography (ECoG) recordings.

Magnetoencephalography is a functional non-invasive neuroimaging technique capable of measuring the weak magnetic fields generated by neural activity with millisecond time resolution. Precisely, it measures the magnetic field outside the head (Baillet 2017). This rich bandwidth enables measuring real-time brain activity that allows studying neural populations dynamically. Unlike the electrical current, the magnetic field passes through the scalp, skull, and brain tissue without any distortion. MEG is a complex signal and requires advanced signal processing and signal extraction expertise to be optimally treated. MEG consists of a whole-head array of multiple independent channels that record the brain's magnetic fields across time from different locations. As a result, these multi-channel measurements contain rich spatial and temporal information about the activity of the underlying neural sources. From a computational perspective extracting this information from such complex spatio-temporal measurements is challenging.

This thesis aimed to decode the kinematics of the hands measured by accelerometers from the MEG recordings of the cortical movement-related activity using deep learning techniques. Results obtained on the primary MEG dataset were validated on the IV Brain–Computer Interface (BCI) Data Competition data set number 4, comprising ECoG measurements of the brain activity in a similar task involving continuous movements.

Although there are multiple potential applications of electromagnetic neuroimaging techniques, typical state-of-the-art decoding applications are restricted to classification problems such as classical BCI, sleep staging, and seizure detection. However, the decoding of hand-movements parameters is a regression task; thus, it is a novel study in the field. Different statistical-based approaches can perform regression

and study the neural activity related to a specific continuous target. For example, Sabbagh and colleagues 2020 listed out different approaches: explicit biophysical source modeling, unsupervised decomposition techniques such as Independent Components Analysis (ICA), supervised spatial filtering techniques such as Source Power Comodulation (SPoC), general-purpose machine learning methods, and Riemannian geometry transformation combined with linear models. More precisely, the supervised spatial filtering techniques SPoC was selected as the principal baseline model. It performs spatial filtering of the MEG data estimating the source space that maximally correlates with the continuous target value.

In recent years, DL has outperformed state-of-the-art statistical approaches in different fields, such as computer vision and speech recognition. Therefore, dealing with complex data structures. In addition, DL is well known for its ability to extract meaningful features by learning a hierarchical representation automatically, releasing the researcher to tedious and time-consuming pre-processing and feature extraction pipelines, which requires, most of the time, high-sector knowledge. MEG data are incredibly complex with a high-dimensional spatio-temporal structure; indeed, DL may deal with this complexity in a better way than other state-of-the-art techniques. However, DL approaches are still novel in the sector. Hence, it remains an open question of whether deep learning presents truly advantages over a more traditional processing pipeline (Roy et al. 2019).

The proposed models are Convolutional Neural Network (CNN) since their ability to extract patterns from the input data hierarchically, as well as, they are particularly effective in decoding data with high spatial information (Khan et al. 2018). Indeed, CNN-based architectures are the most used neural network type in the sector (Roy et al. 2019). Two are the principal model proposed. The first one is called MNet, inspired by Aoe and colleagues 2019. The second one is called SCNN, inspired by Kostas and colleagues 2019. Besides, in addition to the SPoC algorithms, two state-of-the-art CNN-based architectures were evaluated as auxiliary baseline models to investigate further the DL decoding capability performing the thesis tasks.

Aims of the thesis

As introduced before, the main goal of the thesis was to investigate the potential and limitations of deep learning techniques to decode continuous hand-kinematics parameters from the electromagnetic brain activity measured by MEG. The following research questions guided the focus of the research:

- Can deep learning decode continuous hand-kinematics parameters from within-subject MEG measurements?
- How precisely can deep learning decode continuous hand-kinematics parameters

compared to state-of-the-art techniques?

Thesis Structure

This thesis contains six chapters: Introduction, Background, Datasets, Research materials and methods, Results, and Discussion.

The background section will discuss the main state-of-the-art knowledge applied during the thesis. It will firstly give a general review of neuroscience, mainly focusing on introducing the neuroimaging techniques analyzed. Additionally, it will introduce the concept of deep learning, focusing on Convolutional Neural Networks used later as proposed solutions. Finally, it will describe the current DL adoption as a neuroimaging decoding method.

The Datasets section will present the two datasets used during the experiments. It will explain the experiments whereby they were generated, the data structure, and the respective pre-processing pipelines.

The Research materials and methods section will introduce the main research methodologies used during the thesis, describing the implementation tools, the evaluation measures, the baseline models, as well as the proposed solutions.

The Results section will present all the outcomes from the different experiments. It will mainly show the principal within-subject analysis on the MEG datasets. Additionally, it will describe the auxiliary result on the ECoG dataset to validate the proposed approaches.

The Discussion will eventually drive the conclusion by analyzing the obtained results as well as it will propose possible future works.

2 Background

This chapter will give a high-level overview of the thesis background in order to explain the basics of neuroscience and introduce the methods used during the experiments. First, a quick introduction of basic concepts of neuroscience explains how brain activities are generated. After that, an overview of neuroimaging will be described, followed by a more specific description of motor decoding, the main task of the experiment conducted during the thesis. Next, an in-depth overview of MEG explains the potentiality and limitations of the recordings. Additionally, the SPoC algorithms will be explained. Following a general introduction of deep learning will focus on the key architectures and concepts used later on during the experiments. Finally, a brief explanation of the potentiality of the DL-based MEG pipeline will be given.

2.1 Neuroscience

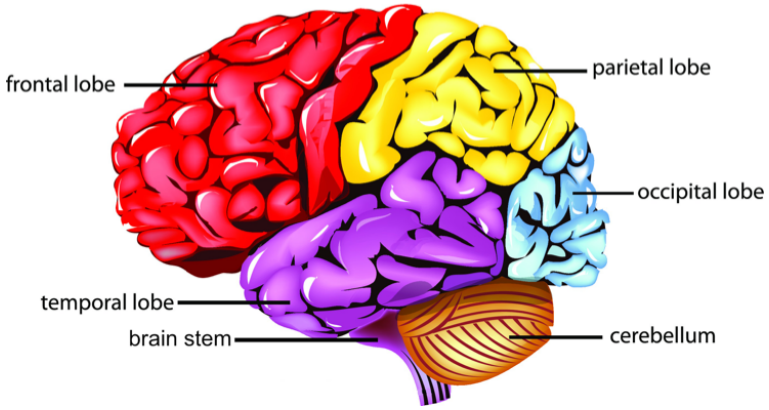
2.1.1 Basic of neuroscience

Neuroscience is the discipline that aims to study the nervous system to analyze its properties, functionality, and malfunction. Various sub-disciplines such as molecular and cellular neuroscience study how neurons process signals physiologically and electrochemically. Cognitive and behavioral neuroscience study the relationship between physiological function and neural activity. Computational neuroscience uses mathematical models to describe the fundamental principles behind the nervous system's cognitive and physical abilities. In recent years, new techniques are emerging, such as neuroimaging, as well as the capability to generate more sophisticated experimental techniques that lead neuroscientists to address clinical and abstract questions.

The nervous system is divided into two classes: the central nervous system (CNS) and the peripheral nervous system (PNS). The CNS consist of the brain and spinal cord where the brain process all the information, and the spinal cord spread out the information and signal to all the body. Generally, each part of the brain is specialized in different functions such as control of movement, memory information, or visual information. The PNS consists of all the other sensory and motor nerves outside the spinal cord and brain. It is the connector between the CNS and the rest of the body, such as organs and limbs.

Brain

The brain is the organ that interprets information from the outside world through our five senses. It controls the body functionalities as well as other phenomenons such as emotion and creativity. As introduced before, the brain is divided into different parts with different functionality. Figure 1 briefly shows this division in 6 different parts: the brain stem stands at the top of the spinal cord. It relays and regulates information between the brain and spinal cord. Then, there is the cerebellum, a wrinkled structure that contains a massive amount of cells, and it has the responsibility to control voluntary movement, specifically when learning something new. Above the brain stem and the cerebellum, there is a wrinkled outer layer of the brain called the cortex. It is split into four different sub-parts called lobes; each one specialized in different functions. The frontal lobe plans for action and controls the movement, among many other things. Behind the frontal lobe, the parietal lobe is a primary sensory area (Bisley 2017), which senses the sensory touch information, taste and temperature. Below the parietal lobe, there is the temporal lobe that receives the auditory information as well as it has an essential part in creating and preserving long-term memory. Finally, the occipital lobe receives and processes all the visual information such as distance, depth perception, and color determination.



Source: Jillian Higgins 2017

Figure 1: Basic brain division in main parts based on different high-level functionality.

Neuron

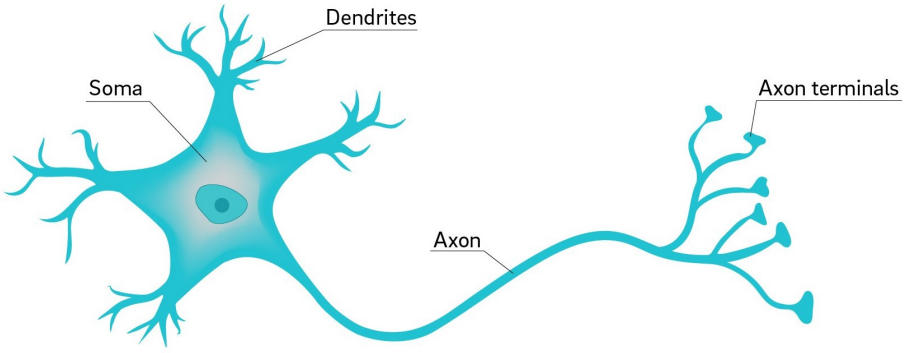
Neurons are the computational units of the brain, and, on average, human brains are formed by around 100 billion neurons, and each of those has about 10000 connections to other neurons called synapses, forming a complex network (Herculano-Houzel 2012). The way neurons communicate with each other is through electrical signals

generated by the ion currents across their cellular membranes.

Figure 2 is a representation of a neuron with its main components. The dendrites are the places where neurons receive information from other cells. In the center of the neuron, there is the soma and the nucleus, which form the body of the neuron, supporting and regulating its activity. Finally, the axon which sends the information to the next cells. The axon is additionally divided into three different regions with different functionalities: the axon hillock, the axon, and the axon terminals. The axon hillock is the region that attaches the axon to the soma. It has a crucial and specific role in deciding when the neuron fires. The axon propagates the signal in the form of action potential to the axon terminals, where the neuron makes synaptic contact with targeted next cells.

To briefly explain the firing process, initially, the neuron maintains a steady potential difference between the intra- and the extra-cellular space, called the resting potential. When neurotransmitters exchange chemical signals to specific receptors, the transmembrane potential changes increasing the number of positive charges inside the nucleus. By the time the number of positive charges exceeds a certain threshold, an action potential will fire, propagating the electrical signal along the axon.

Additionally, it is essential to know that neurons never function isolated; they are organized in neural circuits. Different neural circuits can vary in dimension, characteristics, and direction of the information carried. Synapses interconnect the neurons inside a circuit in order to execute a specific task when activated. The concept of neural circuits and the fact that multiple neurons fire together, generating a chain of multiple simultaneous action potentials, leads to the generation of a measurable current, the base of some neuroimaging techniques.



Source: David Baillot 2018

Figure 2: Representation of a neuron with its main parts.

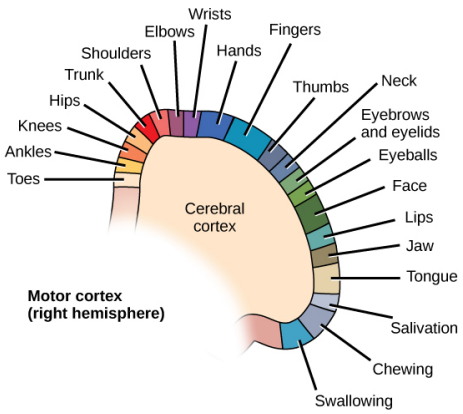
2.1.2 Motor system

Voluntary movements are regulated by the brain and involve cortical, subcortical, and peripheral nerves. To perform coordinated and precise movements, the cooperation of many sensory and motor neurons is required: the primary motor cortex (M1) sends the main movements commands.

Motor decoding

Motor decoding aims to analyze the connection between brain activity and all different body movements, such as hips or hand movements. The brain region involved in planning and controlling voluntary movements is part of the frontal lobe, called the motor cortex. Additionally, this motor cortex can be divided into three regions: the primary motor cortex, the premotor cortex, and the supplementary motor area (SMA). The primary motor cortex works as the main actor in planning the movement, and it is directly connected to the spinal cords through long axons. The premotor cortex regulates the preparation for movement as well as the sensory guidance of movement. The supplementary motor area function as a scheduler of sequences of movement and coordinates movements that include both parts of the body, for instance, left and right hand.

Moreover, keeping in mind that different brain parts are responsible for different functionality, different parts of the motor cortex are responsible for moving different parts of the body, as shown in Figure 3.



Source: CNX OpenStax 2016

Figure 3: Motor cortex division by part of the body controlling area. Additionally, the more the cortical representation is larger, the more precise control of movement has. Therefore, the cortical representation is proportional to the precision of movement performed by the corresponding part of the body.

Various experiments have shown that the modulation of neurons inside the motor cortex was related to many kinematic aspects of movement, such as direction, amplitude, velocity, and acceleration. Moreover, multiple statistical and analytical tools such as multi-regression analysis and information theory have driven to the conclusion that movement parameters are most probable encoded inside groups of neurons instead of in single neurons (Ebner et al. 2009). Additionally, some experiments prove that cortical motor neurons could allow the decoding of a two-dimensional limb movement direction with valuable accuracy during the preparation and execution of movements (Rickert et al. 2009).

Moreover, Kolasinski and colleagues 2019 show that kinematics parameters have specific spatial encoding as well as distinct temporal encoding. Indeed, both: spatial and temporal resolutions contain useful information about the movement.

Decoding hand movements

The hand is one of the freest and versatile parts of the human body, and its adaptability to various tasks makes the hand movement one of the most difficult to encode; hence, the human hand skills are still unsurpassed by any artificial hand. The hand is a motor organ and a sensory organ such that the two functionalities cannot be separated. As a result, the tactile information captured by the hand sensory functionality plays an essential role in the ongoing motor behavior (Benedetti 1994). Two movement features generally describe hand movements: kinematics parameters representing the spatial and motion aspects such as position, velocity, and acceleration and kinetics parameters that focus on describing muscles and forces (Branco et al. 2019). In technical terms, the hand movement features a high number of degrees of freedom (DoF) (Santello et al. 2013). This complexity makes hand control tremendously challenging; therefore, researchers in robotics and neuroprosthetics are continuously exploring new ways of model physical and functional hand characteristics. For example, just thinking about grasping an object, multiple parameters can change, such as the distribution of forces used or the position of the hand itself. These degrees of freedom that allow the hands to fulfill their tasks in different situations come with the price that the CNS has to model a complex system. Moreover, analyzing action, also considering the temporal dynamics and not only a snapshot of a static situation, clearly adds complexity to the decoding task.

BCI and Motor Imagery

In general, motor decoding is studied from a Brain–Computer Interface (BCI) perspective. BCIs are tools used to restore communication between the brain and physical part of the body where the respective neural connections are damaged by trauma, stroke, or neuromuscular disorder that leads to paralysis or loss of motor

function. As a result, the possibility to decode hand kinematics parameters produces control signals that can be potentially used to restore motor function in patients with a neuromuscular disorder.

A comparison between motor imagery and actual motor movement allows a full understanding of the potential of motor encoding and BCI. Motor imagery is the action of imagining a movement of a part of the body while keeping the muscles in a rest position. Multiple studies concluded that: even though brain activity can be registered during motor imagery, actual movement generates higher activation levels, faster responses, and different spatial distribution compared to motor imagery (Batula et al. 2017; Pfurtscheller et al. 1977). As a result, this should be taken into account while designing the BCI system. Although there are some differences between actual movement and imaginary movement, imagery-based BCI applications present a great opportunity for treating neuromuscular disorder patients, such as patients affected by upper motor neuron lesion (Lisi et al. 2017).

2.1.3 Neuronal oscillations

Neural oscillation is a rhythmic electrical activity that involves largely distributed populations of neurons. Oscillations are grouped based on their frequency range and amplitude. There are five main types of brainwaves: alpha, beta, delta, gamma, and theta summarized in Table 1.

Moreover, voluntary movement is related to sensorimotor rhythms (SMR). They are known to be the oscillatory activity occurring in primary somatosensory and motor cortices (Pfurtscheller et al. 1996; Pfurtscheller et al. 1977). Precisely, SMR consists of two distinct frequency components in the alpha and beta bands, respectively, at approximately 10 Hz and 20 Hz (Pfurtscheller et al. 1997).

Wave	**Frequency**
Delta	0.2-3 Hz
Theta	4–7 Hz
Alpha	8–13 Hz
Beta	14–31 Hz
Gamma	32–100 Hz

Table 1: Brain waves main frequency bands.

2.1.4 Neuroimaging

Brain imaging is a broad discipline of neuroscience that uses images to represent and examine brain activity. There are multiple neuroimaging applications, such

as, among others, detecting abnormalities in the brain, direct the treatment plan, evaluate treatment responses, or evaluate disease progression. In the past decades, neuroimaging techniques progress outstandingly; therefore, they gained a central role in clinical diagnosis and biomedical research. Brain images are divided into two broad categories: structural and functional imaging. Structural techniques create snapshots of the brain, and they are generally used to show the structure of the brain to support the diagnosis of some brain diseases such as brain trauma and brain tumors. On the other hand, functional images are consequent snapshots that aim to capture brain functionality while carrying out specific tasks. As a result, they directly contribute to brain understanding since they allow to map and locate specific neural circuitry with specific ongoing brain functions (Bunge et al. 2009).

Moreover, there are substantial differences between different brain images techniques due to diverse characteristics such as signal sources, energy levels, spatial resolution, and temporal resolution. Hence, different techniques end up to provide complementary information. For example, classical MRI images are well-known for their capability to represent soft tissues; however, they display a poorly bone structure. On the other hand, CT imaging, thanks to its effectiveness of display structures according to their density, can provide clear anatomical structure representation. Contrarily, they fail to represent soft tissue. Besides, standard non-invasive functional imaging methods differ for spatial and/or temporal resolutions. For instance, fMRIs are well-known to have a high spatial-resolution. Nonetheless, they have low temporal resolution such that they can effectively access the whole brain at the expense of the loss of neural population dynamic. By contrast, EEG and MEG have a higher temporal resolution that can dynamically analyze the neural population.

Each technique has some specific limitation that drives to noisy information encoded by the images. Hence, the combination of multiple techniques can enhance the overall quality of information monitored (Zhang et al. 2020). In this thesis, two different functional brain imaging techniques will be used: MEG and ECoG.

2.2 Magnetoencephalography (MEG)

This section will focus more on describing the principle and functionality of MEG measurement since it is the neuroimaging technique used during the principal thesis experiments. MEG is a functional non-invasive neuroimaging technique known to have a high temporal resolution. In general, when neurons fire, they generate electrical activity. In particular, electrical activity that goes parallel to the skull can be detected by measuring the out-going magnetic fields. The brain magnetic field strength is about 10^{-15} T (femtotesla), approximately a millionth of the Earth's magnetic field strength. As a result, this magnetic field is a lot smaller than the ambient magnetic noise. Consequently, it is crucial to be able to isolate these tiny brain magnetic fields.

2.2.1 Measurement principle and limitations

The basic principle of MEG relies on the involved technology that senses the neural activity. Each electrical current inside the brain generates magnetic induction that can be measured with sensors placed out of the brain, therefore, remote from the actual source. Specifically, the current is generated by postsynaptic potentials (PSPs) of synchronous activities of cortical pyramidal neurons (Hämäläinen et al. 1993). The general MEG signal band is about 0.5–1.000 Hz, with 1–80 Hz being the most emblematic. Extracranial magnetic inductions are on a scale of 10^{-15} T. Indeed, in order to be able to measure such tiny fields, highly sensible sensors are required (Baillet 2017). The current state-of-the-art pick-up sensors are called SQUIDs (superconducting quantum interference devices). SQUIDs rely on superconductivity principles and properties; they are designed to reduce the impact of external magnetic fields such as radio signals, moving vehicles, and the power-line fields. As a result, SQUIDs are used in MEG implants to compose magnetometer and gradiometer sensors.

Given that magnetic induction travels to the air, MEG sensors are not attached to the skull. Instead, in state-of-the-art devices, they are embedded in a thermally insulated tank, called dewars, filled with 70–100 liters of liquid helium cooled to $-269\,^{\circ}\mathrm{C}$ such that they preserve superconductivity (Baillet 2017; Hämäläinen et al. 1993). All of these described properties of the MEG pick-up system aim to increase the signal-to-noise ratio, therefore, optimizing data quality.

MEG measurements are multiplexed signals represented in different channels frequency; hence, each channel captures the neural activity differently. Each signal depends on the position and location of the pick-up coil with respect to the neural sources. Therefore, MEGs are strongly affected by the subject's head position and the subject's head physics. As a result, these limitations are essential factors to consider during data acquisition and interpretation while comparing subjects and between different experiments.

The magnetic field permittivity is homogeneous across the different components involved: scalp, skull, and the air between the scalp and sensors. As a result, there is a low level of signal distortion between sensors and sources. However, the more the source is more profound in the brain, the more the signal-to-noise ratio decreases. Consequently, there is a belief that MEG cannot detect activity from medial cortical or subcortical brain regions (Baillet 2017).

Although MEG represents a valuable and distinctive technique compared to other functional neuroimaging methods such as fMRI and EEG as well as the recommendation for clinical MEG systems is growing. The high cost of the system is slowing down the clinical adoption of MEG. Thus, there is no broad accessibility to MEG systems by the research community yet (Baillet 2017).

Additionally, another limitation is the complexity of MEG data flow since there is a potentially greater number of sources than the number of pick-up coils. Hence, the channels embed the source model indefinitely, making the source modeling fundamentally ill-posed.

To sum up, MEG advantages consist mainly of the highly-temporal resolution. This reach bandwidth enables measuring real-time brain activity that allows studying neural populations dynamically. Moreover, the non-invasive aspect provides a significant advantage during experiment set-ups, increasing subject comfort while decreasing the preparation-time. On the other hand, MEGs do not have a good spatial resolution, the source model is an ill-posed problem, and eventually, the costs of building a MEG system are still substantially higher compare to some other functional neuroimaging techniques.

2.2.2 Open MEG datasets

Despite the cost-oriented limitation, MEG data are continuously gathering more interest between researchers, such that open-data sharing initiatives are publishing new open MEG dataset. Currently, three are the principal MEG dataset repository available: MEG from the Human Connectome Project, MEG from the Open MEG Archives (OMEGA), and MEG from the CAM-CAN initiative. However, most of these datasets are from resting-state MEG and a lack of standard file format from raw MEG data Baillet 2017.

In past years, one of the main bottlenecks of MEG was the high-density temporal resolution of the recording, which generates a large amount of data difficult to store. However, the recent improvement in storing system leads to more large scale distribution of MEG data. Besides, this high-volume recording opens multiple opportunities for machine learning and big data tools in the sector. Thanks to the capability to represent neurodynamics brain activity, MEG occupies a unique position in the neuroimaging techniques landscape.

2.2.3 Neural oscillation and hand encoding

Brain activity involved in limb kinematics can be detected with functional imaging, although in order to investigate correctly neural mechanisms, high temporal resolution is needed. Thus, MEG temporal resolution potentially represents a valuable feature to study motor activity (Jerbi et al. 2007). A temporal analysis using MEG recording of kinematic together with the decomposition of MEG signal into alpha, beta, and gamma frequencies unveiled specific kinematic encoding across distinct bands. The kinematic showed significant encoding in the alpha band after movement onset and

the beta band before movement onset. On the other hand, gamma bands reveal more information about synergistic muscle activation (Kolasinski et al. 2019).

2.2.4 Comparison with EEG and ECoG

EEG

Electroencephalography is a functional neuroimaging technique that monitors the electrophysiological brain activity. MEG and EEG are frequently considered sister techniques since they measure the electrochemical current flows between brain cells. The following comparison will explain that these techniques can be considered complementary; however, not redundant (Baillet 2017).

First of all, the clear difference is that MEG records magnetic activity, whereas EEG captures electrical activity. A consequence of this, by properties, is that the magnetic field passes through the scalp, skull, and brain tissue without any distortion. Contrarily, the electrical field is affected by distortion while passing the different parts of the brain between sensors and sources. Thus, this is a significant advantage of MEG recordings (Singh 2014) since it leads to a better spatial resolution. Besides, the SNR of MEG can be estimated by empty-room recording and consequently effectively filtered out. Differently, in EEG, actual skin contact is required.

Secondly, EEG electrodes are placed along the scalp. Therefore, it requires additional preparation time to settle the subject. Indeed, preparation time is averagely much shorter during MEG experiments. On the other hand, the standardization of electrode montage in EEG reduces recording variability due to head position and head physics, allowing better comparison between different experiments and between different subjects. In this matter, MEG recordings are strongly dependent on the position of the pick-up coil with respect to the different neural sources.

Thirdly, EEG systems are cheaper since MEG has higher initial and maintenance costs. However, some cost-effective MEG solutions are coming up in the market, generating greater MEG affordability as research tools (Baillet 2017).

Fourthly, the EEG is historically the oldest functional technique. Indeed, the research productivity of EEG is in volume incomparable with MEG. However, research publications regarding MEG are constantly increasing in the past 25 years (Baillet 2017).

To sum up, despite the differences pointed out before, the millisecond temporal resolution allows both the techniques to measure neural population dynamics in real-time such that they can be considered similar to each other. Hence, due to the lack of scientific paper on MEG in the contest of deep learning, during the thesis,

not only the MEG literature was reviewed but also the one of EEG.

ECoG

Electrocorticography or intracranial electroencephalography (ECoG) is the invasive version of EEG, where the electrodes are placed directly on the exposed surface of the brain. It is therefore required a surgical incision to implant the electrode grid. The closeness of the electrodes to the neural sources increases the quality of the recordings compared to EEG. As a result, ECoG provides both: high spatial (mm scale) and temporal (ms scales) resolution. Additionally, another advantage of ECoG over MEG and EEG is that recordings inside the brain are less susceptible to artificial distortion. The high-temporal resolution of ECoG captures the neural motor dynamics as in MEG. Therefore, ECoG can be used for motor decoding. Precisely, ECoG measurements compose the second dataset used to validate the proposed model.

2.2.5 MEG pipeline for regression tasks

The main state of the art approaches used in MEG are generally based on a statistical pipeline, and they can differ task by task. Statistical learning generally relies on covariance matrices estimation (Sabbagh et al. 2019). First of all, MEG signals are usually pre-processed using some general steps such as downsampling, band-pass filtering, and windowing. Additionally, specific types of well-known noises are normally filtered out utilizing specific artifact removals techniques such as ocular noise. However, in general, some high-level knowledge of the sector is required to apply artifact removal techniques. Finally, hand-feature engineering extraction enhances the data relevance for future processing (Roy et al. 2019). The ultimate goal of the pre-process pipeline is to increase the SNR enhancing the data quality accordingly.

After pre-processing MEG signals, different statistical and machine learning approaches are used in the literature to perform regression tasks. Following a brief classification of the main methodologies:

- Explicit biophysical source modeling commonly uses anatomical constraints to infer the most likely source configuration. This method uses penalized linear inverse models that estimate the current source distribution. The well-defined forward model is used in order to integrate the prior anatomical knowledge in the process such that the signal space is considered as a linear estimation of the source space, and therefore the inverse model can be thought of as biophysical spatial filtering (Sabbagh et al. 2020).

- Unsupervised decomposition techniques such as Independent Components Analysis (ICA) decomposes the signal space into a new feature space such that the new features are mutually statistically independent. As a result, these new components can be considered as statistical sources linearly correlated by a mixing matrix called the lead field matrix. This technique can be thought of as generating unsupervised spatial filtering of the source space.

- Supervised spatial filtering methods have recently become popular. They not only estimate the source space but also the estimation is based on criteria relevant to a specific continuous target. Some of these techniques are Common Spatial Pattern and Source Power Comodulation (SPoc). The latter is used as a baseline model during the thesis experiments.

- General-purpose machine learning techniques, including logistic regression, linear discriminant analysis, and linear support vector machines, analyze the sensor spaces directly without considering any data generating mechanism.

- A more recent approach consists of applying transformations in the Riemannian geometry and use linear models to learn from spatially correlated power-spectra. These methods have set a new-state-of-the-art in multiple classification tasks.

2.3 Source Power comodulation

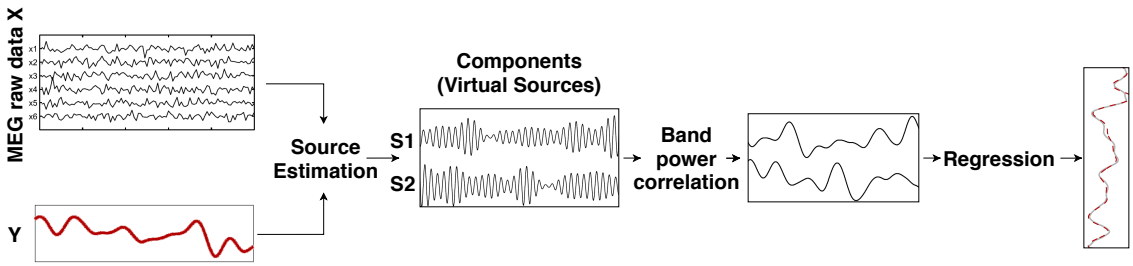
SPoC is a supervised spatial filtering technique that performs spatial filtering that maximally correlates with a continuous target. It has been discovered firstly by Dähne and colleagues 2014. Additionally, there are no limitations of the target variable (following denoted by y), which can be any scalar function of time, such as hand kinematics parameters.

SPoC uses the target variable during the decomposition process to prioritize those components whose power comodulates with the respective target variable. SPoC generally investigates MEG spectral power and the target y in order to estimate a functional relationship between stimulus and neural amplitude modulation. Specifically, the main experiments of the thesis investigate the neural modulation with respect to hand kinematics.

There are two main ideas to the base of the SPoC approach: first, the decomposition of the multi-channel MEG data into a new space of virtual source components. Second, guide the decomposition using the target y variable. The final result of this process is a set of spatial filters W, which intrinsically optimizes the comodulation between the MEG signal and the variable y. The concept of the spatial filter in literature commonly refers to a vector of weights meant to extract the signal to one source while suppressing the activity of the other sources. As a result, the signal generated is a close approximation of the underlying source activity.

In a more practical point of view the SPoC algorithm generates the filter $\boldsymbol{W}$ to maximize the covariance between y and the power of the filtered signals. This spatial covariance matrix of the source signal $\mathbf{W}^T x(t)$ is easily obtained from $\mathbf{W}^T \mathbf{C}_i \mathbf{W}$, therefore, the spatial filters $\mathbf{W}$ are generated by solving a genearalized eigenvalue decomposition problem (Sabbagh et al. 2020). For example: in the study of Dähne and colleagues 2014 two are the algorithm proposed for possible SPoC implementations; they differ in the definition of comodulation. On the one hand, one approach optimizes the source power correlation between the y value and the predicted value of $\tilde{y}$. On the other hand, the second approach optimizes the source power covariance between the y value and $\tilde{y}$.

To conclude, according to (Sabbagh et al. 2020), SPoC algorithms outperform other standard techniques such as ICA to recover the association between motor variables and neural activity. Hence, SPoC has been selected as the principal baseline model. Moreover, even though the SPoC approach is originally intended to be used with MEG and EEG data, it is not limited to these techniques. Therefore, it can be readily used with functional invasive techniques such as ECoG.



Source: Dähne et al. 2014)

Figure 4: Flowchart of the SPoC algorithm. In the left the input data are shown: example of MEG recording (top-left plot), y target value (bottom-left plot). The two type of inputs are combined to do a correlated source estimation. First, the algorithm generates the new virtual sources (center-left plot). Secondly, the band power of this sources is correlated with y generating the new correlated components (center-right plot). Finally, a regression algorithm is trained to optimally approximate the y expected value (left plot).

2.4 Deep learning

Artificial Intelligence (AI) is a broad branch of computer science that aims to create systems that can function intelligently and independently, replicating human behaviors. The sub-branch that uses data to perform tasks and answer questions is called Machine Learning (ML). ML provides systems the ability to automatically learn and improve without actually being programmed. This process is called training, whereby the model learns from the available data to automatically perform some

actions. ML systems are currently in all the devices around us doing various actions from recommending the new song while listening to music to helping doctors diagnose cancer.

Deep learning is a subset of Machine Learning, which is inspired by the functionality of the brain. It aims to extract patterns from data using a particular model called Neural Network. The main difference between ML and DL is that, on the one hand, before fitting machine learning algorithms, programmers apply a process of hand-engineered features extraction in order to narrow down the features space. On the other hand, in deep learning, this process is mainly performed directly on raw data by the training process of the model. As a result, DL algorithms will automatically learn a hierarchical representation of the features providing a more end-to-end approach.

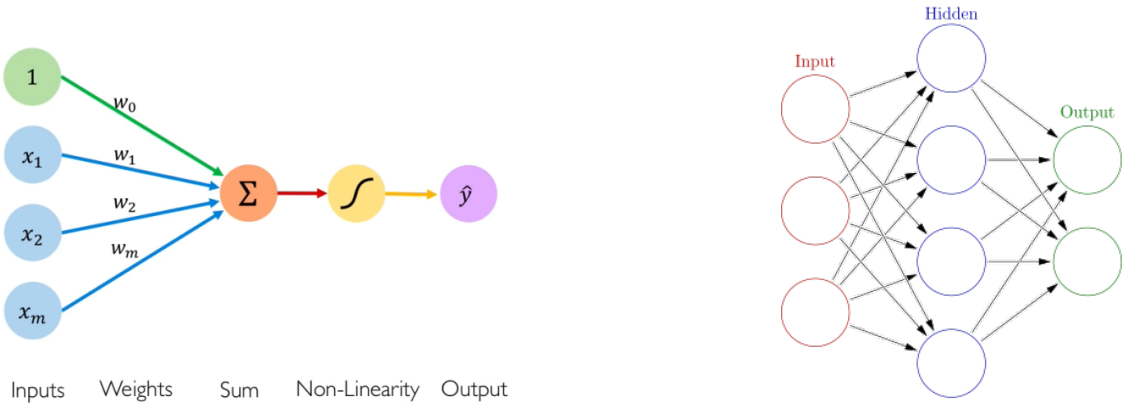
However, this automated learning process comes at a cost. Firstly, DL requires more data compare to ML that implies more storage required as well. Secondly, DL requires more computational power and, finally, more sophisticated software to build models. Despite DL algorithms have existed for decades, DL is only recently massively adopted since these requirements are getting more readily affordable due to the rise of big data, modern GPUs are getting cheaper as well as open toolboxes, such as Tensorflow (Abadi et al. 2016) and Pytorch (Paszke et al. 2017) have become extremely streamlined.

2.4.1 Neural network in general

Artificial neural networks (ANN or NN) form the base concept of DL. NNs are computing systems inspired by biological neural networks of human and animal brains. NNs take data as input and train themselves to recognize patterns inside the data, learning to predict a specific output based on the input data. The fundamental building block of NN is called neuron, and it is inspired by the functionality of a neuron inside the brain. Multiple of these neurons are combined to generate a layer of the network, and, eventually, multiple layers can be combined to generate the final structure of a NN. The first building block created is the Perceptron (Rosenblatt 1958), the main idea behind is called forward propagation whereby a vector of inputs $\mathbf{X}$ is correspondingly multiplied with a vector of weights $\mathbf{W}$, sum all the results together, and, finally, the summation passes through a non-linear activation function $g(\cdot)$ that eventually produces the final output y. Additionally, an additional term called bias w_0 can shift the activation function to the left or the right. The mathematical formula of the Perceptron is the following:

$$\tilde{y} = g(w_0\, \mathbf{X}^T\, \mathbf{W}) \quad \text{where:} \quad \mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$$

Figure 5a shows the basic structure of the Perceptron.



(a) **(b)**

Source: b) Glosser.ca 2013

Figure 5: Graphical representation of the basic structure of the Perceptron (a) and a basic NN (b).

Neurons can be combined to generate layers of the network, building a complex hierarchical model. These layers are commonly called hidden layers and they are generally connected with all the input values or with all the output values of the previous layer. If so, the layer is considered fully-connected. As a result, a basic neural network architecture consists of the input layer, hidden layers, and a final output layer, as in Figure 5b.

Training a neural network is solving an optimization problem. Each neuron connection has an arbitrary weight assigned to it, therefore, during training, these weights are continuously updated to reach an optimal value in terms of minimizing or maximizing the objective function. This optimization strictly depends on the optimizer used, such as Stochastic Gradient Descend (Kiefer et al. 1952). For example, in a minimization problem, the optimizer aims to minimize a given loss function. In other words, the optimizer will assign the weights to make this loss function as close to 0 as possible. This process is executed multiple times, fitting the model with the input data repeatedly until specific stopping criteria are met. As a result, the model will continuously learn and update the model weights accordingly.

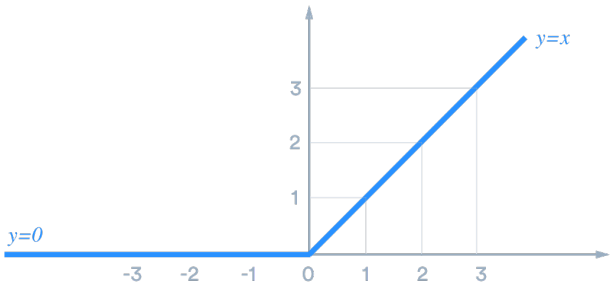
The learning process is done by passing the whole data multiple times through the model; a single pass of all the data is called an epoch. At the beginning of the training process, during the model initialization, the weights are set arbitrarily. Once the loss is computed, the model will calculate the gradient of the loss for each weight. After that, the learning rate, generally a small number that decides how much the weight can vary, is multiplied. Finally, each weight is updated with respect to its gradient.

The gradient of each weight is calculated using the concept of backpropagation. The intuition behind the backpropagation is that starting from the comparison between outputs nodes with respect to the actual expected output, the optimizer understands if the weights should increase or decrease and it updates the weights accordingly going backward through the network.

Activation function

The activation function is biologically inspired by activity in our brains, where neurons fire or are activated by different stimuli. Generally, given a specific stimulus, some neurons are fired, and some are not. As a result, the activation function, in a basic case, gives a number close to 0 to neurons that are not relevant to the final tasks. Thus, they are "not activated". On the other hand, it gives a value close to 1 to relevant neurons, hence, they are "activated". To remark that not in all the cases, the two boundaries are 0 and 1. The purposes of activation functions are to introduce non-linearities into the network that is one of the main advantages of neural networks since input data, in real life, are most of the time non-linear. Hence, non-linearity activation functions allow approximating arbitrary complex functions better compared to simple linear activation functions.

An example of one of the most widely used activation function is the Rectified Linear Unit (ReLU) (Nair et al. 2010) (Figure 6) that transform the input to the maximum of either 0 or the input itself. As a result, the idea behind ReLU is that the more positive the neuron, the more activated it is.



Source: Liu 2017

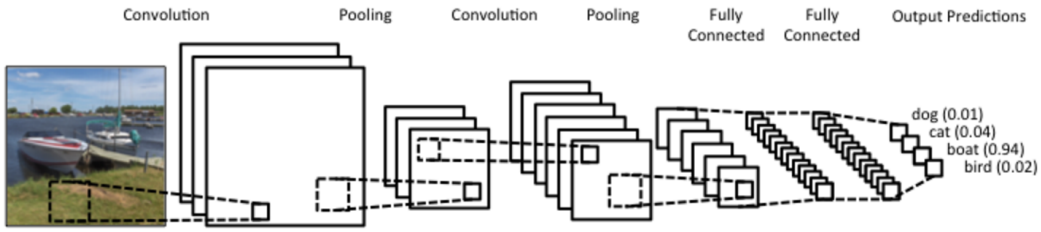
Figure 6: Plot of the rectifier linear unit.

2.4.2 Convolutional neural network (CNN)

The convolutional neural network is a particular type of NN that is mainly known to be particularly effective in analyzing data with high spatial information such as images (Khan et al. 2018). However, CNNs can optimally analyze other complex

data structures, such as time series. We can think about CNNs as a particular ANN that is capable of discovering or detect patterns. The central concept behind CNN is a linear mathematical operation between matrices called convolution, from which CNN derives its name. The building block of CNN are filters used to extract the relevant features from the input data applying the convolution. This filter progressively generates a features map of the input that encode different relevant features. A graphical representation of a CNN consists of steps in which features have been progressively extracted starting from a complex input. The increasing number of convolution operations applied to the input leads to a simple and easily recognizable set of output features. Therefore CNNs follow a sequential architecture, typically non-cyclical.

In general, a CNN is composed of the input layer, one or more convolutional layers, one or more fully-connected layers, and, finally, the output layer. Non-linearity is added as in ANN throughout non-linear activation functions. Figure 7 is an example of a basic CNN. Besides recognizing spatial patterns inside data, CNN has other main advantages: the weights in a convolution layer are shared between neurons. Therefore, the final number of parameters drastically decreases, making the analysis of complex structured data such as MEG lighter and faster. Additionally, the structure of CNN can be easily parallelized. In the next sections, a more in-depth explanation of the convolution operation and the main building blocks of CNN, such as convolutional layer and pooling.



Source: Cvlr 2017

Figure 7: Representation of the architecture of a general CNN. The input data passes by different layers in which convolutional operations or dimensionality reduction operations are applied to extract meaningful features. Finally, a fully connected layer predicts the targetted class.

Convolution operation

Convolution is a function that takes two different functions f and g as input and combines them into a third different function $(f \circledast g)$. The convolution has different algebraic properties such as commutativity, associativity, and distributivity, among

others. In Figure 8 the formula and a graphical example of convolution:

$$f \circledast g =\equiv \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

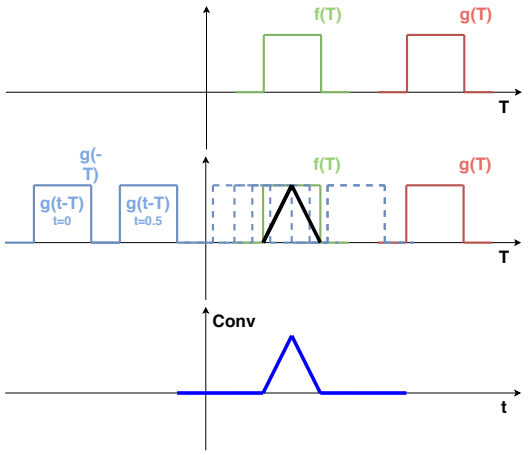


Figure 8: Mathematical equation and example of the convolution operation. In the right figure is shown: (top) two example function $f(\tau)$ (green) and $g(\tau)$ (red). (middle) a simulation of the convolution process, whereby the intuition behind $f(\tau) \circledast g(\tau)$ is that taking $g(-\tau)$ (blue) which is the reflection of $g(\tau)$ along the vertical axis and sweep it accordingly with time t until it overlaps the function $f(\tau)$. During these steps, the two functions are multiplied and eventually integrated. As a result, since the integral of zero from negative infinity to infinity is zero, the convolution will remain zero as long as f, and g do not overlap. On the other hand, when the two functions will start to overlap, the integral will be a positive number that will increase until the two functions completely overlap and decrease to 0 afterward. (bottom) the resulting convolution $f(\tau) \circledast g(\tau)$.

Convolution Layer

The convolutional layer is the backbone of CNNs. Likewise, the Perceptron, the convolutional layer, receives inputs, transforms the input, and sends the output to the next layer. However, the transformation applied is different. Inside the convolutional layers is where the network extracts spatial patterns from the input data through multiple convolutional operations between given filters (feature detectors) and the input data. The number of filters that each layer has is set in the beginning to determine the number of different transformations that will be applied to the input data generating a set of feature maps that represent the distinct features extracted.

Max Pooling

Max pooling is a type of operation that is usually added to a CNN following an individual convolutional layer. Max pooling mainly reduces the dimensionality of

the data so that the more activated neuron are preserved. Therefore, it adds a small amount of translation invariance. Additionally, it reduces the number of parameters reducing computational load as well as it generally reduces overfitting. In Figure 9 a concrete example of max-pooling.

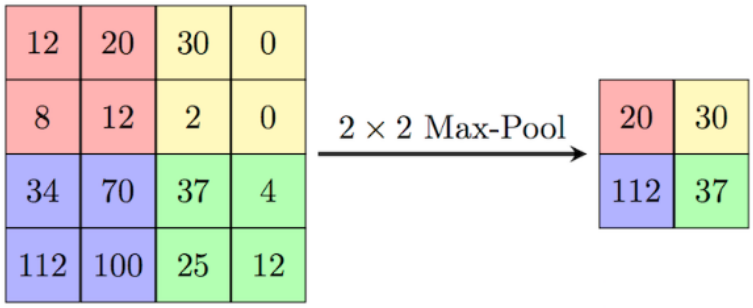


Figure 9: Example of max-pooling in action. It generally works by taking the input neurons in small blocks and keep only the max value per block.

2.5 Deep learning with MEG/EEG recordings

Deep learning is currently outperforming state-of-the-art techniques in multiple fields thanks to its capability to automatically learn meaningful features directly from data. As a result, also in neuroscience, researchers are developing new DL-based approaches to solve different problems. MEG data are incredibly complex with a high-dimensional spatio-temporal structure; therefore, DL may deal with this complexity in a better way than other state-of-the-art approaches. Despite new approaches to MEG signal analysis have recently emerged by using machine learning techniques and the amount of open MEG data available is constantly increasing (Baillet 2017), DL approaches are still novel. Therefore, it remains an open question of whether deep learning presents truly advantages than a more traditional processing approach (Roy et al. 2019).

Advantages and disadvantages

Deep learning is mainly used for its capability to learn complex feature representation directly from raw data, avoiding the tedious process of extracting hand-engineered features before training the model, such as in a general ML pipeline. This advantage is very relevant in MEG signal analysis since MEG recordings have a low SNR. As a result, the general feature extraction process is generally very tedious as well as it requires domain-specific knowledge. Therefore, a DL end-to-end approach can save researchers a considerable amount of time. Additionally, the hand-engineered feature extraction process depends on the state-of-the-art understanding of the data

generation process. Hence, DL may extract more effective or expressive features. However, DL models mostly function as black-boxes. Thus, they are generally limited in terms of neurophysiological interpretability. Moreover, deep learning models are data hungrier. Therefore, they require a large amount of input data.

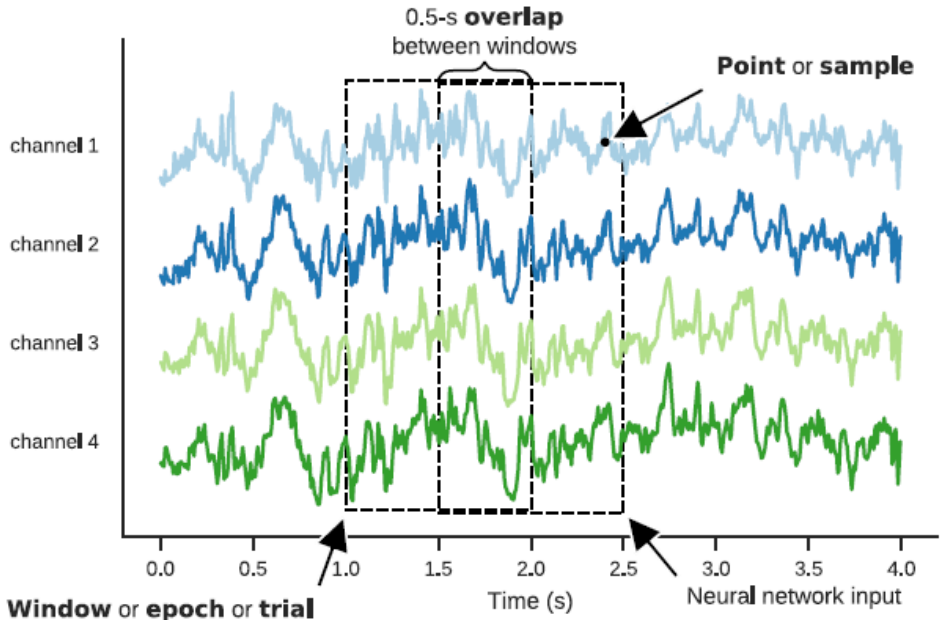
Overview of architecture used

Different kind of neural network architecture has been used in DL-based MEG/EEG pipelines. The most used architectures in the literature are CNNs. Another type of architecture used is the recurrent neural network (RNN), usually used to process sequential data and, therefore, capable of extracting temporal features. Moreover, in recent years, the number of studies that adopt a combination of CNNs and RNNs are steadily increasing. Additionally, besides the supervised methods also autoencoder (AEs) have been used to learn a new representation of the input data given some particular constraints (Roy et al. 2019).

Windowing

A general step of processing MEG and similar multivariate time series data in order to fed a DL model is to windowing the signal creating different samples of fixed time length from the original time points. If the MEG recording is of shape $[n\_channel, n\_times]$ such that $X_i \in \mathbb{R}^{C \times T}$ and the window duration corresponds to n time sample. The windowed MEG signals are of shape $[n\_epochs, n\_channel, n]$ where $n\_epochs = \frac{n\_times}{n}$. For example, if X_i has shape $[204, 100000]$ and $n = 1000$. The epoched data matrix will have the shape of $[100, 204, 1000]$.

Additionally, DL MEG pipelines usually use overlapping windows. This concept generates windows that share some time points. For example, if the duration of the window is 1 second and the overlap is 0.5 seconds. The first window generated contains the time point from 0 to 1 second, the second window from 0.5 to 1.5 seconds, etc. In the epoched MEG signals, each sample is used as a single entry of the DL model. Therefore, the concept of overlapping windows is used to generate a greater number of samples compared to non-overlapping windows, hence augmenting the input data. Figure 10 graphically explains the windowing system as well as introduces some typical terms.



Source: Roy et al. 2019

Figure 10: example of windowing. The channels are split along the time domain into smaller segments called windows. These new sub-segments generated are the input of the neural network. Additionally, the two windows (dashed squares) are overlapped to generate overlapping windows. The window generated can also be called epochs or trials. Each time point is usually called a point or sample.

To remark, there are some possible ambiguities between MEG processing terms and general deep learning terms. Table 2 aim to disambiguate these common terms.

Term	Neuroimaging	Deep learning
Epoch	Equal-length spans of data extracted from raw original data.	A single pass of all the data to the model during training.
Sample	Single time point of a signal.	Single entry of a dataset, it generally corresponds to a raw of a dataset.

Table 2: Disambiguate common terms between neuroimaging and DL.

3 Datasets

This section describes the dataset used during the experiments. The thesis experiments mainly analyze a MEG motor dataset collected in Aalto university during a hand motor experiment, following referred to as the MEG dataset. Secondly, to prove the model developed using the primary dataset, an open dataset of ECoG of the 4th Brain–Computer Interface Data Competition is used, following referred to as the ECoG dataset. Following a detailed explanation of the recording procedures, the data structure, and specific data processing applied.

3.1 MEG dataset

Measurement procedure

The measurement consisted of a MEG-based method which gave real-time feedback about the brain function of the subject. Experiments were conducted to develop new approaches to motor rehabilitation of patients suffering from hemiparesis, weakness of one part of the body caused by different medical conditions, such as stroke. However, at the beginning of the study, data of healthy patients were gathered while performing hand movements. These healthy subjects data are the only ones used in the thesis experiments.

MEG was recorded with a 306-channel Elekta Neuromag Vectorview (MEGIN Oy, Helsinki, Finland) in a high-end 3-layered magnetically shielded room at the MEG Core of Aalto Neuroimaging, Aalto University. Moreover, five small head position indicator coils were attached to the head of the subjects. Additionally, small accelerometers were attached to the back of both hands.

During the measurement, the task of the subject was to roll a softball with a circular motion of the left or the right hand in the direction (clockwise or counter-clockwise) shown on the screen (Figure 11). The accelerometer output was given on the screen as feedback as a horizontal bar extending to the right or left such that the subject was able to monitor its own hand movement. Each trial lasted for 20 seconds and was followed by a 10-second break. During the break, the hand was kept still. During each of the three experimental sessions, the subject performed 20 repetitions, resulting in a total of 60 20-second trials. Overall, the experiment lasted for roughly 30 minutes. All the data were sampled at 1000 Hz.

The final dataset consisted of MEG recordings capturing the brain activities and the accelerometer data capturing the hand-movement parameters used as the target variable. The task of the thesis was to decode the target variable from the MEG measurements.

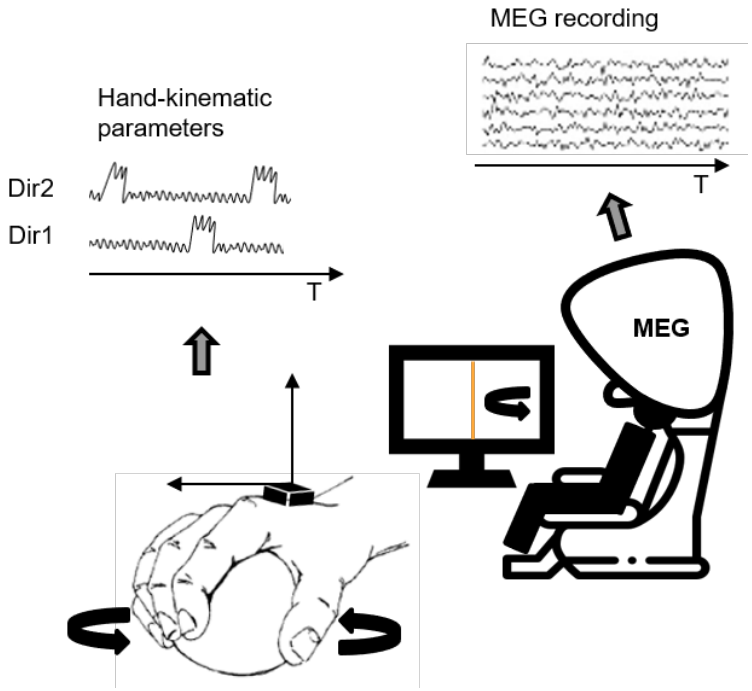


Figure 11: Summary of the experiment. The Subject was placed in the MEG device that was measuring the brain activity. Meanwhile, the accelerometers were capturing the movement of the hands. The different cues were given on a screen using an arrow. The location of the cue arrow (left or right side) told which hand the subjects had to move, and the direction of the arrow told whether the subject has to rotate the ball clockwise or counterclockwise. For example, in the figure scenario, the subject had to move the right hand counterclockwise. Finally, a horizontal bar extending to the right or left was given as real-time feedback, such that the subject could monitor its own hand movement. Each accelerometer was measuring two directions.

Pre-processing of the MEG signal

The first step required to be able to process the three different runs together was applying Maxfilter (Taulu et al. 2004); a spatial signal space separation algorithm to remove external noise as well as realign the head position moving the data to a standard space (e.g., between different runs or different subjects).

Before windowing the data, the line currents noise frequencies 50 and 100 Hz was notch filtered out. Moreover, MEG data were band-pass to 1–70 Hz. Next, the channels were scaled by subtracting the mean and dividing by the standard deviation across all time points and epochs. As a result, all channels are normalized to have a mean of zero and a standard deviation of one. Besides, the channels were downsampled to 500 Hz.

The resulting normalized input data matrix is $X \in \mathbb{R}^{E \times C \times T}$, where $C = 204$ are the gradiometers channels; the number of epoch E and time points T depend on the window duration. As mentioned before, the experiments consist of predicting hand-kinematic parameters from the recorded brain oscillations. Thus, the solution should be able to predict the parameter with a specific prediction rate such that the results prove some concrete potential for possible future applications. Hence, the fixed prediction rate is 200 ms. Besides, this prefixed rate standardized the output such that it help during the comparison between the provided solutions as well as during the model tuning process. In other words, the models should predict the target variable each 200 ms. Therefore, the window duration and the window overlap were selected accordingly (i.e., duration = 1 sec, overlap = 0.8 sec) such that each epoch could have been used to predict one target value.

Pre-processing of the kinematic signal

Accelerometer data consisted of continuous measurements of acceleration along two different axes for both hands such that the target variable Y had four channels. Likewise, the MEG recording the accelerometers data were downsampled by a factor of two, getting to a sampling rate of 500 Hz.

First of all, the neural activity spatially differs between the two hands. Thus, the models are designed to predict the movement parameter for one hand only, such that to predict both hands parameters, two different models are required. After this initial splitting, the acceleration along the two axes are combined to get one hand parameter to simplify the decoding process. PCA was used to extract the combined parameters such that the new parameter is a linear combination of the two-axis that maximize the variation in the data. Moreover, PCA can be inverted to transform the data into its original space. Finally, the target parameter was aggregated to generate one value per epoch, thus, having one value each 200 ms to predict. This aggregation was done by summing the absolute values of the instant accelerations. Following a recap of the y pre-processing steps:

1. y_{raw} is composed of four channels C encoding hands rotation as acceleration values with two different axes per hand. $y_{raw} \in \mathbb{R}^{C \times T}$, for example C = 4, T = 400000.
2. The two hands are treated separately. $y_{left} \in \mathbb{R}^{C' \times T}$, with, C' = 2, T = 400000.
3. The two axes are combined using PCA. $y_{pca} \in \mathbb{R}^{1 \times T}$, with T = 400000.
4. The y values are reshaped summing the values of 200 ms chunks such that each epoch has one value. $y \in \mathbb{R}^{E \times 1}$, with, E = T / 200 = 4000.

Data structure

The resulting normalized input data matrix is $X \in \mathbb{R}^{E \times C \times T}$, where $C = 204$ are the gradiometers channels; the number of epoch E and time points T depend on the window duration. As mentioned before, the experiments consist of predicting hand-kinematic parameters from the recorded brain oscillations. Thus, the solution should be able to predict the parameter with a specific prediction rate such that the results prove some concrete potential for possible future applications. Hence, the fixed prediction rate is 200 ms. Besides, this prefixed rate standardized the output such that it help during the comparison between the provided solutions as well as during the model tuning process. In other words, the models should predict the target variable each 200 ms. Therefore, the window duration and the window overlap were selected accordingly such that each epoch could have been used to predict one target value. Specifically, a window duration of 1, 1.2, and 1.4 seconds were tested during the experiments. Therefore, T were 501, 601, and 701, respectively, since the data were previously downsampled to 500 Hz. Figure 3.3 summarize the data structure eventually used.

3.2 ECoG dataset

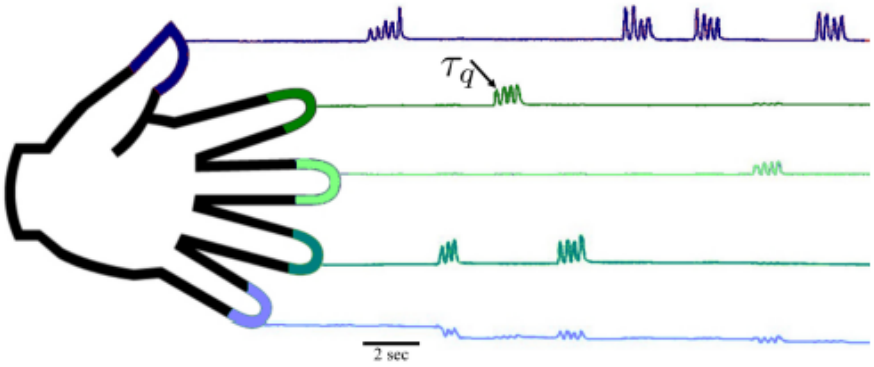
IV Brain–Computer Interface Data Competition data set number 4 (Schalk et al. 2007), composed of ECoG recording of brain activity of finger movements. Here to download the dataset if needed.

Measurement Procedure

The experiments to record this dataset was carried out at Harborview Hospital in Seattle, Washington. Recordings of three different epileptic patients compose this dataset. The task during the experiment consisted of a BCI application that, providing visual stimuli, was acquiring brain activity and the flexion of individual fingers using a data glove. Therefore, the BCI application stored the brain signal and separately the flexion of each finger in an additional data file. Each subject had subdural electrode arrays containing 48–65 platinum electrodes. As a result, the ECoG recording signals consist of 62, 48, and 64 channels from subjects 1, 2, and 3, respectively.

More specifically, each subject was cued to move a particular finger by displaying on a computer monitor the corresponding word such as thumb or little. The screen display was alternating between cue and blank screen with a step of two seconds. Each subject averagely moved the displayed finger three to five times during each cue, varying across subjects and fingers. In total, 30 movement stimuli have been

cued for each finger randomly displayed. The experiment lasted 10 minutes for each subject. The signals were additionally band-pass filtered between 0.15 to 200 Hz and sampled at 1000 Hz (Tangermann et al. 2012). In Figure 12 an example of the finger recording.



Source: Miller et al. 2008

Figure 12: Example of data captured by the data gloves. The experiment continuously captures the fingers flexions. Additionally, the figure shows a correlation between some finger movements, such as while moving the little or the middle; also, the index is automatically and involuntarily moved.

ECoG data pre-processing

The data were already band-pass filtered between 0.15 and 200 Hz by the authors. The 50 and 100 Hz frequencies were notch filtered out, and the signals were down-sampled by a factor of two. Next, the channels were standardized to have zero mean and a standard deviation of one. The windowing duration and overlap were generated as in the MEG dataset to guarantee a prediction rate of 200 ms.

Pre-processing of the kinematic signal

The target value of the decoding problem consists of finger flexion. The models were designed to predict the flexion of a single finger only. Additionally, as in the MEG dataset, the y value was reshaped accordingly to the prediction rate of 200 ms, taking the mean of the vertical position in the 200 ms.

3.3 Datasets comparison

Even though the two datasets are similar regarding the fact that they contain the recording of brain activity while the subjects were performing hand-movement tasks, they substantially differ in some aspects. First of all, in general, ECoG has a higher SNR compere to MEG. Hence, the data are less noisy. Second, the dataset size is smaller in the ECoG dataset than the MEG one in terms of the number of subjects available and total minutes of recording per each subject. Table 3 summarize the input data size of two subjects, one per dataset. The MEG recordings are approximately 30 minutes for each subject. On the other hand, the ECoG recordings are approximately 7 minutes for each subject. As a result, due to the experiment are within-subject, the recording size used to train the models is six times bigger in the MEG experiments than the ECoG experiments. Finally, the MEG recording subjects were healthy; contrarily, the ECoG recordings were from epileptic patients.

Duration-Overlap	**MEG data (ExCxT)**	**MEG recording (min)**	**ECoG data (ExCxT)**	**ECoG recording (min)**
1.0-0.8	[11330, 204, 501]	30	[1996, 62, 501]	7
1.2-1.0	[11327, 204, 601]	30	[1995, 62, 601]	7
1.4-1.0	[11324, 204, 701]	30	[1994, 62, 701]	7

Table 3: Comparison between MEG and ECoG datasets.

4 Research materials and methods

This section explains the research methodology used during the thesis, as well as describes the main experiments conducted. It begins with an overview of the methodology and with the formalization of the problem in technical terms. After that, a section describes the implementation tools and the general workflow of the analysis. Finally, the proposed solutions are introduced along with the baseline models.

4.1 Research methodology

The final goal of this thesis is to understand the potential of deep learning in decoding motor-related brain activity and further compare the results with state-of-the-art techniques. Specifically, the decoding is a regression task; therefore, the solution should predict a continuous value. As already mentioned previously, the decoding of continuous parameters using MEG is still a novel topic in the field. As a result, different baseline models were selected to benchmark the performances of the proposed models. The principal baseline model was the supervised spatial filtering method, Source Power Comodulation (SPoC). This novel method outperformed other classic state-of-the-art techniques (Sabbagh et al. 2020) in a similar task to the one under analysis. As a result, SPoC provides a reasonable solution as a baseline model. Moreover, some DL state-of-the-art models were selected. Specifically, LeNet-like and ResNet-like architectures were selected since they were designed to work with complex data structures such as images. Thanks to these baseline models, the potentiality of general architectures was further investigated. Additionally, given that the relative power spectrum was integrated into the deep learning proposed solutions described later, a combination with the relative power spectrum and a simple Multi-layer Perceptron was evaluated to complete the analysis.

Finally, two different CNN-based models are proposed as solutions after a careful review of the literature. Even though they were proposed in classification tasks, the intuitions behind them were promising, as well as they were previously designed and tested on MEG recording.

4.2 Evaluation measure

The main measure used to evaluate the solutions is the Root Mean Square Error (RMSE). The RMSE calculates the error or difference between the predicted value $\tilde{y}$

and the true values y. Following the mathematical formula:

$$\textbf{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y - \tilde{y}\right)^2}$$

The RMSE is a non-negative scale-dependent measure of the accuracy of the model prediction. The lower the RMSE, the better the model predicts. Since the RMSE is the square root of the average of squared errors, the effect of each error is proportional to the size of the error itself; thus, RMSE is sensitive to outliers.

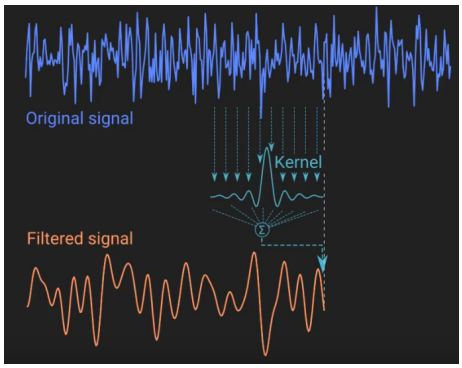
4.3 Formalizing the problem in a technical way

Each channel can be seen as a time-varying linear-combination of the activity of the underlying sources. The brain is a complex system; therefore, multiple sources are generally operating simultaneously. Thus, each channel records a mixture of the underlying sources along with the different noises, such as measurement and environmental noise. As a result, each channel encodes the activity of the underlying sources differently, such that there is a spatial correlation between channels. The high-resolution of the MEG and ECoG encodes dynamics of neural activity; thus, encoding both spatial and temporal information. Hence, both the dimension may contain useful information for the decoding of the analyzed task. To analyze the recording, generally, two different types of filters can be applied: spatial and temporal filters.

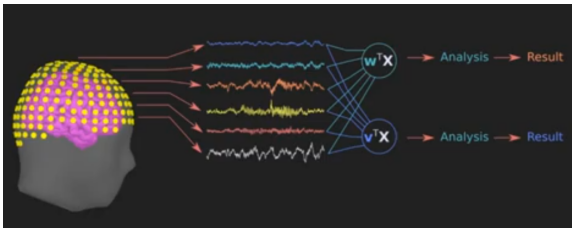
Temporal and Spatial filters

Temporal filters are applied to the time dimension of MEG signals such that the signal is combined with some specific filter to optimize statistical characteristics generating a new filtered signal. The filter function that is used to weight the time points is called the kernel.

Spatial filters consist of a weighted sum on the channel dimension of the MEG recording meant to extract the signal to one source while suppressing the activity of the other sources. Ideally, the different new virtual sources generated are a close approximation of the original source space.



(a) Temporal Filter



(b) Spatial Filter

Source: Mike X Cohen 2019

Figure 13: Conceptual representation of a temporal filter (a) and a spatial filter (b). (a) shows the process of temporal filtering, which combines the original signal with a specific kernel multiplying each time point with the respective kernel point and generates the new filtered signal. (b) shows ideal spatial filtering in which the different channels are combined together each time point to generate the virtual sources. As a result, temporal filters are computing over time in one channel. On the other hand, spatial filters are computed over space for each time points.

To sum up, the MEG and ECoG recordings are multi-channel signals with spatial and temporal correlations. Additionally, MEG usually has high SNR; therefore, the DL model must learn from a noisy and complex data structure.

4.4 Implementation tools

This section will introduce the main tools used. The implementation was done in Python (Van Rossum et al. 2009) and the main libraries used are MNE (Gramfort et al. 2014; Gramfort et al. 2013) and PyTorch (Paszke et al. 2017).

- **MNE** is an open-source python package for exploring, visualizing, and analyzing human neurophysiological data, such as MEG, EEG, and ECoG. It was mainly used to perform classical MEG pre-processing and split the input signal into epochs. Besides, MNE already provides a SPoC implementation.
- **PyTorch** is an end-to-end machine learning open-source framework optimized for computing via GPU. It was used to build and train the proposed DL architectures.
- **MLflow** (Chen et al. 2020) is an open-source platform for the machine learning lifecycle. Specifically, it has been used as a tracker of all the different

experiments conducted as well as to track the tuning processes. It provides an effective and structured way to log parameters, metrics, and models.

- **Triton** is the Aalto high-performance computing cluster. It has been used to boost the computation parallelizing the different models' training processes.

4.5 Baseline models

This section will introduce the baseline models selected, explaining the reason for the selection, the general workflow, and the final implementation. Four baseline models were selected.

4.5.1 SPoC

The SPoC algorithms provided by the MNE package is an implementation of the work by Dähne and colleagues 2014. The model allows performing source power comodulation extracting a specific number of orthogonal spatial filters that maximally correlate with a continuous target variable. The new expected number of components has to be given a priori; thus, the model cannot automatically understand the optimal number of components. As a result, the optimal number of components was calculated using cross-validation. MNE APIs are inspired by Scikit-learn API (Pedregosa et al. 2011). Therefore each MNE estimator can be treated as a standard sklearn estimator such that a sklearn pipeline can easily be used to train the optimal number of components. Specifically, 10-fold cross-validation with a penalized linear model (Ridge Hoerl et al. 1970) was used to predict the continuous value and find the optimal number of spatial filters.

Cross-validation is a model validation technique. It allows comparing different models to understand which is the best one and which is the best combination of model parameters to the specific task. In machine learning, there are two steps to understand the validity of a model. First, the model is trained, and, second, the model is tested to a hold-out subset of the dataset not used during training. Cross-validation splits the dataset in k sub-sets. Then it trains the model using $k-1$ subsets and tests it on the last hold-out subset. It performs this process since all the k subsets are used as the test set, keeping track of results of each model that are eventually averaged across the k-trials. During the experiments, a 10(k)-fold cross-validation was used.

Ridge regression is a penalized linear model. It stabilizes the learning process since it helps against over-fitting the training data, introducing a small amount of bias into how the linear model fits the training data, therefore, dropping the residual variance that allows better prediction of unseen data. In the experiment pipeline,

ridge regression is used to estimate the target values from the newly generated components such that the new spatial filters generated can be validated.

During cross-validation, two parameters were tuned: the number of components of SPoC and the Alpha parameter of the ridge regression implemented in Scikit-learn. The latter is the regularization strength. It is used to reduces the variance of the estimates. The larger the value, the stronger the regularization. Table 4 shows the hyper-parameters searching space.

	Source Space	**Sub 8 best model**
# of components	[2–30]	16
Alpha	[0.8, 1.0, 2, 5, 10]	2
Duration and overlap (sec)	(1.0, 0.8), (1.2, 1.0), (1.4, 1.2)	(1.4, 1.2)

Table 4: SPoC hyper-parameter searching space.

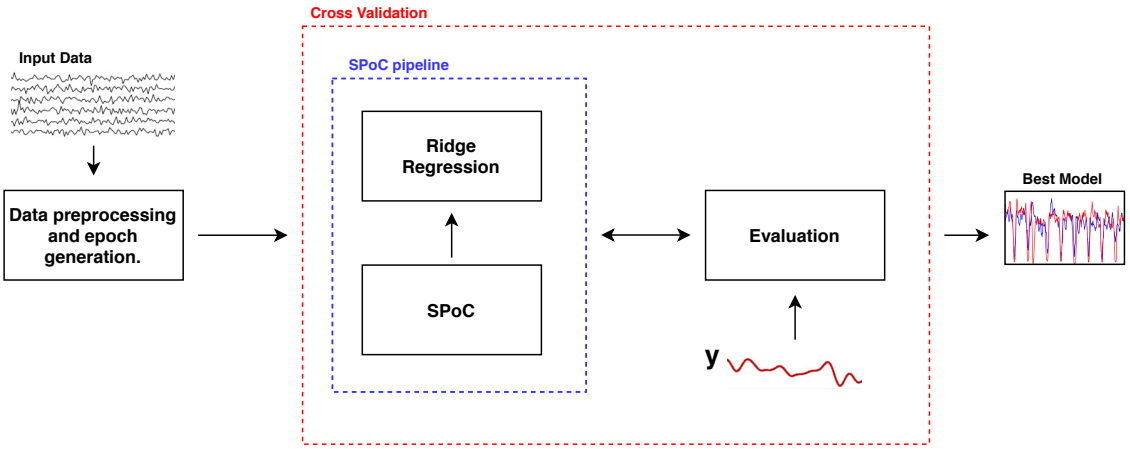


Figure 14: SPoC workflow. First, the raw input measurement are pre-processed and epoched such that they are ready to be processed by the SPoC algorithm. The raw data $X \in \mathbb{R}^{C \times T}$, where C is the number of channels and T is the number of time points corresponding to the whole recordings. They are transformed by windowing to $X \in \mathbb{R}^{E \times C \times N}$, where E is the number of epoch and N is the length of the chosen window. Additionally, the y target value is transformed in order to have one target value per epoch; hence, $y \in \mathbb{R}^{E}$. Then, the decoding SPoC pipeline (blue dashed rectangle) decodes the epoched input. The SPoC algorithm generates new components that the Ridge regression uses to predict the expected values. Moreover, cross-validation (red dashed rectangle) is used to validate the decoding pipeline, specifically, to select the optimal number of new components that the SPoC algorithms generate, and, finally, the best model is selected among the different trials.

4.5.2 RPS and MLP

This model was implemented to check the proprieties of the RPS to extract meaningful features from the input data. The RPS was combined with an MLP to add non-linearity and increase the capability of approximate the target variable.

The RPS was implemented in 4 steps:

1. Compute the modified periodogram using Welch methods (Welch 1967) to get the power spectral density.
2. Calculate the average band power approximating using the composite Simpson's rule to get it for a specific target band.
3. Divide the average band power of the specific target band by the total power of the signal to get the relative power spectrum.

The band powers extracted were 6 (Table 5), and the RPS was calculated per each channel each epoch. As a result, there was one RPS value per channel, for example, in the primary experiment in which the data input had 204 channels. The RPS was generating [204, 6] RPS values. Figure 15 summarize the model pipeline.

Wave	Frequanecy
Delta	1–4 Hz
Theta	4–8 Hz
low-Alpha	8–10 Hz
high-Alpha	10–13 Hz
Beta	13–30 Hz
low-Gamma	30–50 Hz

Table 5: Brain waves main frequency bands.

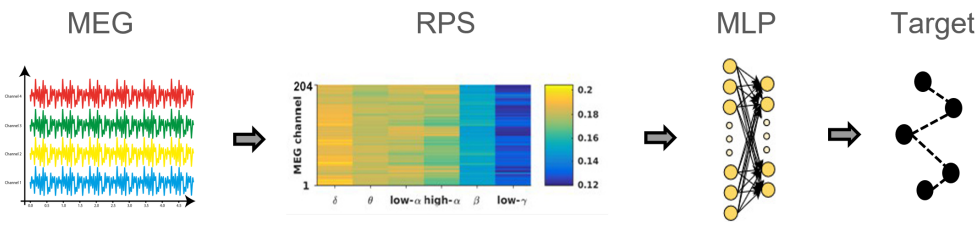


Figure 15: RPS and MLP ideal pipeline representation. The pre-processed MEG data are used to extract the RPS values. After that, the RPS values were fed as input to a MLP to eventually predict the target values.

Parameter	Source Space	Sub 8 best model
Batch size	80, 100, 120	100
Learning rate	[3e-3 , 1e-4]	0.002
Duration and overlap (sec)	(1.0, 0.8), (1.2, 1.0), (1.4, 1.2)	1.4-1.2
Activation Function	ReLU	ReLU
Optimizer	Adam	Adam

Table 6: Summary of the RPS-MLP hyper-parameters searching space.

4.5.3 LeNet-5

LeNet-5 (LeCun et al. 1989) is a simple convolutional neural network well known in the literature since it is one of the oldest and successful CNN models. It was initially developed for the classification of images. It has been selected mainly for its simplicity since the network is composed of fewer layers and small kernels dimensions. As a result, the model represents light and fast solution to evaluate whether a relatively simple and swallow network can deal with the complexity of the MEG recordings. The implementation of this baseline model almost followed the original. The implemented LeNet had some little changes, such as increasing the number of filters inside the convolutional layers and increasing the number of neurons inside the FC layers. They were changed to give the network more parameters to learn to boost the performances. Besides, ReLU was used as an activation function and Adam (Kingma et al. 2017) as the optimizer.

Parameter	Source Space	Sub 8 best model
Batch size	80, 100, 120	80
Learning rate	[3e-3 , 1e-4]	0.0013
Duration and overlap (sec)	(1.0, 0.8), (1.2, 1.0), (1.4, 1.2)	1.4-1.2
Activation Function	ReLU	ReLU
Optimizer	Adam	Adam

Table 7: Summary of the LeNet-like architecture hyper-parameters searching space.

4.5.4 ResNet

ResNet (Residual Neural Network) is a well-known architecture based on building blocks that implement the concept of skip connection (He et al. 2015). This network is mainly famous since it can significantly be deeper and still be able to generalize well. Therefore, it has been selected as a baseline model to evaluate the effectiveness

of a complex and deeper network to deal with the complexity of the MEG recordings. Besides, ReLU was used as an activation function and Adam as the optimizer.

Parameter	Source Space	Sub 8 best model
Batch size	80, 100, 120	80
Learning rate	[3e-3 , 1e-4]	0.0011
Duration and overlap (sec)	(1.0, 0.8), (1.2, 1.0), (1.4, 1.2)	1.2-1.0
Activation Function	ReLU	ReLU
Optimizer	Adam	Adam

Table 8: Summary of the ResNet-like network hyper-parameters searching space.

4.6 Proposed models

This section will introduce the two principal CNN-based models proposed as solutions to the decoding task. The first model is called MNet, and the second model is called SCNN. They differ from each other in how they are designed to extract the features from the MEG recordings.

4.6.1 MNet

The main idea of this proposed model is to firstly extract global features from the recording convolving in both spatial and temporal dimensions. As a second step, it aims to extract frequency-based features from the previous global features extracted. Finally, it aims to extract local features. Additionally, MNet integrates the input data with specific band-powers of the MEG signals. Therefore, the input data were processed separately in two different ways and eventually concatenated to be further processed together. Two different versions were tested: MNet without RPS integration and RPS-MNet with RPS integration (Aoe et al. 2019).

Architecture

From a high-level perspective, the MNet is composed of a 2-block CNN, where the input data are processed by a global-filtering block followed by a local-filtering block and an FCFFNN. The input of the network is $X \in \mathbb{R}^{E \times C \times T}$ with E the number of epochs, C the number of channels, and T the time points per each epoch. In the following explanation, the E dimension is not considered assuming that one epoch

corresponds to one data sample fed to the network. Figures 16 and 17 represent the architecture of MNet and RPS-MNet respectively.

Global-filtering Block. The first block of the networks consists of two convolutional layers that aim to extract global features of the input data progressively. Specifically, the first layer is designed with a large kernel that simultaneously applies transformations to channel and time dimensions. This layer is designed to map the input data from $\mathbb{R}^{C\times T}$ to $\mathbb{R}^{G\times 1\times T^{'}}$ with $T' < T$ and G the number of global filters. Hence, collapsing the channel dimension. Moreover, another convolution layer on the time domain was applied to extract frequency features as well as increase the number of transformation applied such that the new data are from $\mathbb{R}^{G'\times 1\times T^{''}}$. Finally, the max-pooling is applied to downsample the data across the time domain.

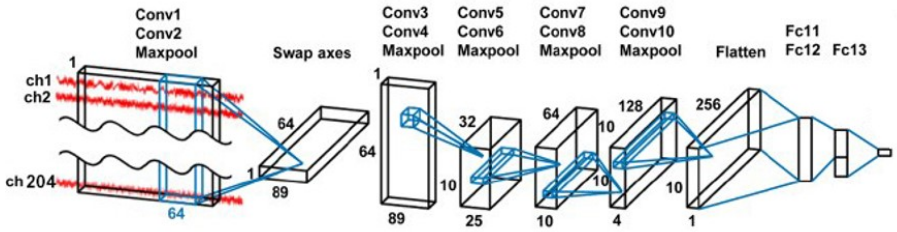
Swapping axes. After the first block, the global filters dimension and the collapsed channel dimension are swapped such that the data are mapped from $\mathbb{R}^{G'\times 1\times T^{''}}$ to $\mathbb{R}^{1\times G'\times T^{''}}$. In this way, the data can be treated like an image changing the direction of the convolution in order to convolve in both times and across the global feature extracted (Tokozume et al. 2017).

Local-filtering Block. This block is composed of multiple convolution layers with small kernels that aim to extract local time-frequency components from the global features extracted previously. This blocks function as mapping from $\mathbb{R}^{1\times G'\times T^{''}}$ to $\mathbb{R}^{1\times G^{final}\times T^{final}}$ applying multiple small convolution to the data. Max pooling is additionally applied after each convolutional layer.

Relative power Spectrum. In the RPS-MNet version only, in order to augment the data fed to the network, the input data were separately transformed to acquire powers of specific frequency bands for each channel as explained in Section 4.5.2. This process was transforming the input data from $\mathbb{R}^{C\times T}$ to $\mathbb{R}^{C\times B}$ where B is the number of frequency bands. The RPS was calculated ahead of the training process and directly passed to the network together with the input data. Therefore, the final RPS matrix $\in \mathbb{R}^{E\times C\times B}$.

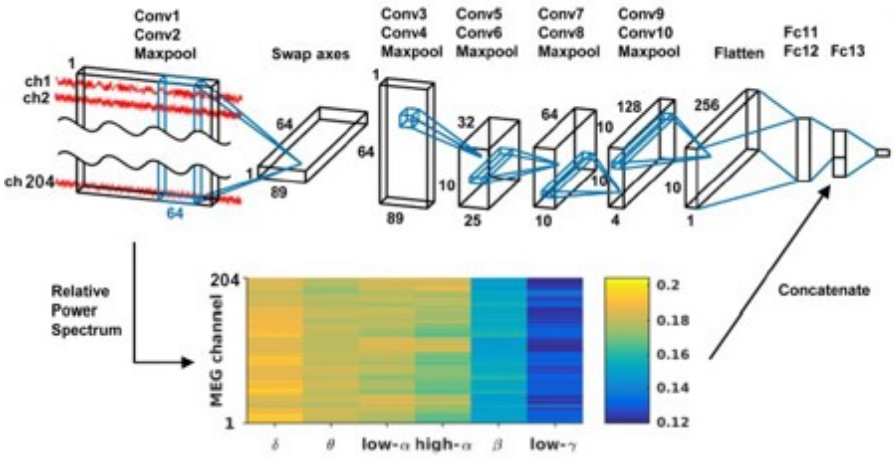
Concatenation. In the RPS-MNet version only, the two different outputs were flattened and concatenated to form the feature space in input to the final FCFFNN block.

FCFFNN. This Block is composed of consequent fully-connected linear layers that aim to combine the previously generated feature by the two different processes in RPS-MNet or by the CNN-blocks only in MNet to predict the continuous target value. Additionally, regularization is introduced after each linear layer using dropout (Srivastava et al. 2014) as well as batch-normalization, thus preventing overfitting.



Source: Aoe et al. 2019

Figure 16: Mnet architecture. The first block composed of Conv1 and Conv2 extract global features, completely collapsing the channel dimension such that it extracts new global frequency features. The second block is composed of Conv3 till Conv10 extracts local features. Finally, the features are processed by fully connected linear layers.



Source: Aoe et al. 2019

Figure 17: RPS-Mnet architecture. The first block composed of Conv1 and Conv2 extract global features, completely collapsing the channel dimension such that it extracts new global frequency features. The second block is composed of Conv3 till Conv10 extracts local features. At the same time, the RPS features are generated from the input data. Finally, the features are concatenated and further processed by fully connected linear layers.

Model Training

The datasets were firstly split into train, test, and validation datasets, respectively the 70%, 15%, and 15% of the input dataset. The training dataset was utilized to feed the network during the training process. The validation dataset was used to evaluate each model during training, calculating the validation loss. Finally, the held-out test set was used to evaluate the trained model.

SGD was used as the optimizer, and the mean squared error function was minimized during training. Moreover, early stopping was adopted as the main stopping criteria of the process. In addition, weight decay of 0.0005 was applied as well as dropout and batch normalization layers after the fully connected ones to avoid overfitting. The RPS values were concatenated directly during the forward step. The extracted bands in which was acquired powers are six ($\delta : 1-4\text{Hz}; \theta : 4-8\text{Hz}; \text{low-}\alpha : 8-10\text{Hz}; \text{high-}\alpha : 10-13\text{Hz}; \beta : 13-30\text{Hz}; \text{low-}\gamma : 30-50\text{Hz}$); hence, the concatenated values are $C \times B$. For instance, during MEG experiments, the concatenated values were 204×6. The hyper-parameters tuning process was computed using a random search technique that automatically generates a combination of the tunable parameters. Random search techniques empirically and theoretically lead to better results compared to a standard grid search algorithm (Bergstra et al. 2012). Table 10 shows the hyper-parameter searching space. Some of the parameters were previously carefully investigated, such as the learning rate to narrow down the respective searching space. The duration and overlap parameters that form the windowing augmentation system are selected such that the epochs generated can predict with the desired 200 ms prediction rate.

As already mentioned before, the model was designed and trained using Pytorch, the tuning of the different parameters was parallelized on the Triton cluster, and the tuning process was monitored using MLflow.

Parameter	Source Space	MNet	RPS-MNet
Batch size	80, 100, 120	80	80
Learning rate	[3e-3 , 1e-4]	0.0016	0.0005
Duration - overlap (sec)	(1.0, 0.8), (1.2, 1.0), (1.4, 1.2)	1.4-1.2	1.4-1.2
Activation Function	ReLU	ReLU	RelU
Optimizer	Adam	Adam	Adam
Weight decay	0.0005	0.0005	0.0005

Table 10: Summary of MNet and RPS-MNet hyper-parameters searching space. In addition to the searching space, the Subject 8 best model hyper-parameter combination is reported.

Layer	Ksize	Stride	Features map
Input			
Conv1	(204, 64)	(1, 2)	32
Conv2	(1, 16)	(1, 2)	64
Pool2	(1, 2)	(1, 2)	
Swap axes			
Conv3	(8, 8)	(1, 1)	32
Conv4	(8, 8)	(1, 1)	32
Pool4	(5, 3)	(1, 2)	
Conv5	(1, 4)	(1, 1)	64
Conv6	(1, 4)	(1, 1)	64
Pool6	(1, 2)	(1, 2)	
Conv7	(1, 2)	(1, 1)	128
Conv8	(1, 2)	(1, 1)	128
Pool8	(1, 2)	(1, 2)	
Conv9	(1, 2)	(1, 1)	256
Fc10	—	—	1024
Fc11	—	—	1024

Table 9: Summary of the MNet architecture.

4.6.2 SCNN

The main idea behind the SCNN is that the features are extracted convolving across the channel domain and the temporal domain separately. Thus, the network applies spatial filtering from the input data convolving on the channel domain only. Eventually, it performs temporal filtering convolving on the temporal domain only (Kostas et al. 2019).

Architecture

From a high-level perspective, the SCNN is composed of a first spatial-filtering block, a second temporal-filtering block, and a final fully-connected feed-forward neural network that combines the extracted features. The input of the network is $X \in \mathbb{R}^{E \times C \times T}$ with E the number of epochs, C the number of channels, and T the time points per each epoch. In the following explanation, the E dimension is not considered assuming that one epoch corresponds to one data sample fed to the network.

Spatial-filtering Block. This block is composed of stacked convolutional layers that aim to extract spatial filters from the input data progressively. This extraction is done by isolating the convolution operation on the channel domain using specific kernel sizes. In other words, the kernels filters are applied over channels for each time point. The layers are designed to reduce the spatial dimension and ultimately collapse it such that the spatial block is a mapping from $\mathbb{R}^{1 \times C \times T}$ to $\mathbb{R}^{S \times 1 \times T}$ where S is the number of spatial filters generated. Namely, a set of spatial transformations are applied to the input data. The hyper-parameters search selected the kernel sizes and the number of layers. Batch-normalization (Ioffe et al. 2015) was applied to make the network faster and more stable.

Temporal-filtering Block. This block is composed of stacked convolutional layers and pooling layers that aim to progressively extract temporal filter from the data output of the spatial block. The filters are designed only to perform convolution across the time domain, such that the kernels are applied over time for each spatial filter previously generated. This block maps the data from $\mathbb{R}^{S \times 1 \times T}$ to $\mathbb{R}^{F \times S \times T^{'}}$, where F are the number of temporal filtering applied to the data. Additionally, max-pooling layers are applied after blocks of two consecutive convolutional layers to down-sample the data. Besides, batch-normalization layers are applied before each pooling layer. A hyper-parameter search selected the number of convolutional layers and the different kernel sizes.

FCFFNN Block. This Block is composed of consequent fully-connected linear layers that aim to combine the previously independent set of spatio-temporal features in order to predict the continuous output value optimally. Additionally, regularization is introduced after each linear layer using dropout; thus preventing overfitting. This block maps the input $\mathbb{R}^{F \times S \times T^{'}}$ form the temporal-filtering block to $\mathbb{R}^{1}$. Hence, predicting one value each epoch.

Moreover, no bias terms were used in each convolutional layers since they are redundant with batch-normalization. The activation function was applied after each layer. The hyper-parameter search has selected the activation function. Figure 18 show the general architecture of the SCNN.

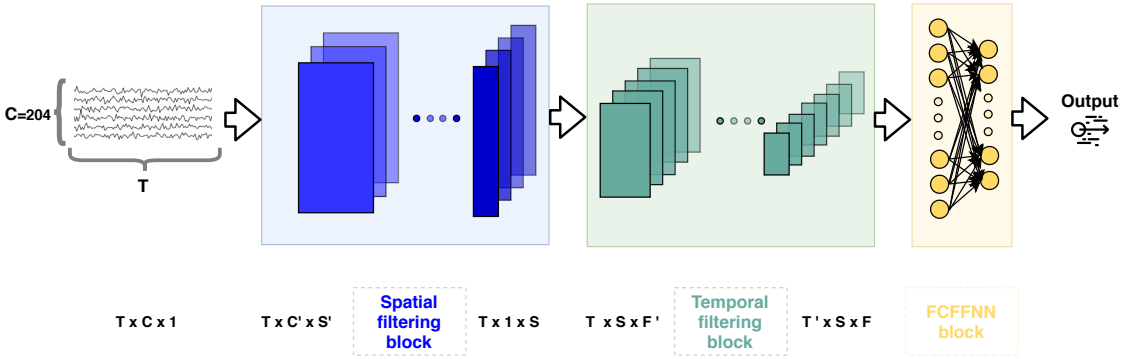


Figure 18: SCNN Architecture. The spatial filtering block (blue) takes in input the epoched recording and generate a new set of spatial filters S while collapsing the channel dimension. A hyper-parameter search selects the number of convolution layer and each kernel dimension. The temporal filtering block (green) takes the output of the previous blocks and performs convolution along the time dimension only, generating T temporal filters. A hyper-parameter search selects the number of convolution layers and each kernel dimension. Finally, an FCFFNN block (yellow) combines the previously extracted spatial and temporal features and predicts the target value.

Relative Power Spectrum integration

In order to increase the performance, the relative power spectrum was integrated during the model training to augment the information feed to the model in the same way as in RPS-MNet, generating a new version of the model following referred to as RPS-SCNN. Figure 19 shows the architecture with the integration.

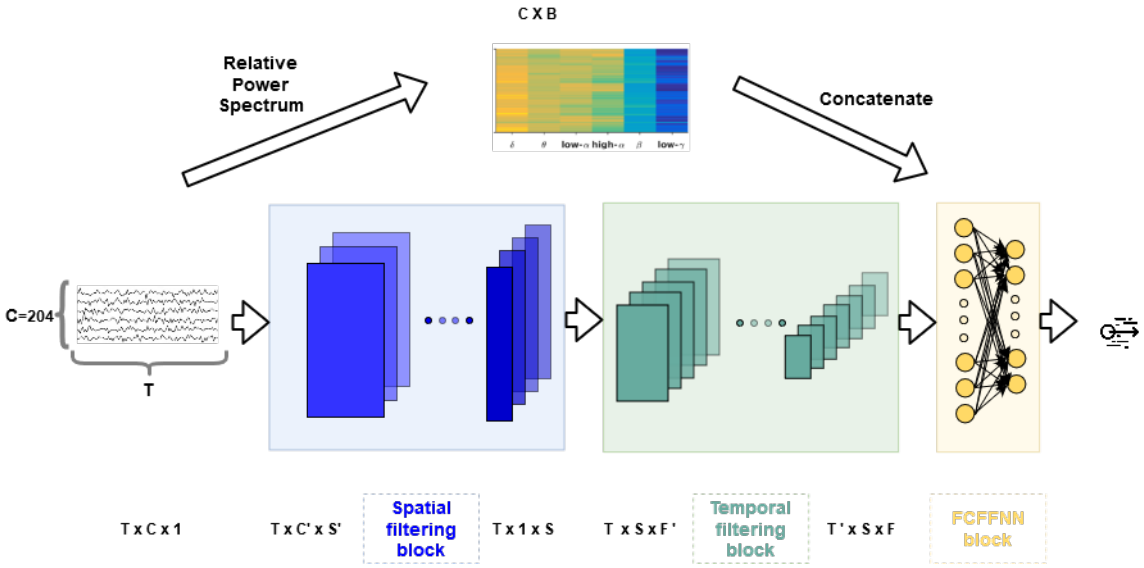


Figure 19: RPS-SCNN Architecture. The spatial filtering block (blue) takes in input the epoched recording and generate a new set of spatial filters S while collapsing the channel dimension. A hyper-parameter search selects the number of convolution layer and each kernel dimension. The temporal filtering block (green) takes the output of the previous blocks and performs convolution along the time dimension only, generating T temporal filters. A hyper-parameter search selects the number of convolution layer and each kernel dimension. Additionally, the RPS features are generated from the input data and concatenated with the network's output features space. Finally, an FCFFNN block (yellow) combines the spatial features, temporal features, and the RPS values previously extracted and predict the target value.

Model training

The datasets were firstly split into train, test, and validation datasets, respectively the 70%, 15%, and 15% of the input dataset. The training dataset was utilized to feed the network during the training process. The validation dataset was used to evaluate each model during training, calculating the validation loss. Finally, the held-out test set was used to evaluate the trained model.

Adam or Stochastic Gradient Descend was selected as the optimizer, and the mean squared error loss function was minimized as the objective function. Moreover, early stopping was adopted as the main stopping criteria of the process. Early stopping is considered as a regularization technique since it prevents overfitting. The activation function was selected between the Rectifier Linear Unit (Relu) (Nair et al. 2010), The Exponential Linear Unit (ELU) (Clevert et al. 2016), and the Scaled ELU (Selu) (Klambauer et al. 2017). To further regularize the training, L2-normalization was applied to all the trainable weights. All the layers were batch-normalized after the

Parameter	Source Space	SCNN	RPS-SCNN
Batch size	80, 100, 120	120	100
Learning rate	[3e-3 , 1e-4]	0.003	0.0019
Duration and overlap (sec)	(1.0, 0.8), (1.2, 1.0), (1.4, 1.2)	1.2-1.0	1.2-1.0
Spatial-filtering kernel sizes	204, [54, 51, 51, 51], [104, 101], [154, 51], [104, 51, 51]	[104, 51, 51]	[54, 51, 51, 51]
Temporal-filtering kernel size	[20, 10, 10, 8, 5], [16, 8, 5, 5], [10, 10, 10, 10], [100, 75], [250]	[20, 10, 10, 8, 5]	[100, 75]
FCFFNN number layers	2, 3, 4	4	3
FCFFNN hidden channels	1024, 516, 248	248	248
Dropout	0.2, 0.3, 0.4, 0.5	0.3	0.3
Activation Function	ReLU, ELU, SELU	SELU	ELU
Optimizer	Adam, SGD	Adam	Adam

Table 11: Summary of the hyper-parameters searching space. In addition to the searching space, the Subject 8 best model hyper-parameter combination is reported.

activation, and dropout was applied after each FC layer.

The hyper-parameters tuning process was computed using a random search technique as in Section 4.6.1. However, here, both: architecture parameters such as the number of layers inside the spatial-filtering block and external parameters such as learning rate were tuned together, generating the hyper-parameters search space summarized in Table 11. Some of the parameters were previously carefully investigated, such as the learning rate to narrow down the respective searching space. The duration and overlap parameters that form the windowing augmentation system are selected such that the epochs generated can predict with the desired 200 ms prediction rate.

As already mentioned before, the model was designed and trained using Pytorch, the tuning of the different parameters was parallelized on the Triton cluster, and the tuning process was monitored using MLflow.

5 Results

In this section, the main results of the thesis will be introduced and explained. The section will firstly show the results using the primary MEG dataset, and, therefore, it will introduce the auxiliary results on the ECoG dataset. The performances of different models were compared using the RMSE metric. All the models have previously experimented on Subject 8. Once all the models were tested on Subject 8, the proposed solutions were validated throughout all the available subjects. Specifically, additional tests were performed for the best model RPS-MNet, the second proposed solution RPS-SCNN and the two main baseline models; the SPoC algorithm and RPS-MLP. All the experiments were within-subject. Indeed, for each subject, each model was trained and tested on its recording only. Additionally, fixing the prediction rate at 200 ms allowed comparing the different tested models with each other fairly.

5.1 MEG experiments

5.1.1 Focus on a sample Subject

SPoC

The SPoC baseline model was implemented as explained in the methodology Section 4.5.1. After the cross-validation process, the best model was selected for each hand. The final RMSE of the best pipeline was 0.976 for the left hand and 0.965 for the right hand. The model is followed referred during the cross-evaluation as SPoC.

Relative Power Spectrum and MLP

The baseline model composed of the relative power spectrum and a Multi-Layer Perceptron was implemented as explained in the methodology Section 4.5.2. The final RMSE of the best models was 0.779 for the left hand and 0.782 for the right hand. The model is followed referred during the cross-evaluation as RPS-MLP.

LeNet

The LeNet-5-like baseline model was implemented as explained in the methodology Section 4.5.3. The final RMSE of the best model was 0.975 for the left hand and 0.978 for the right hand. The model is followed referred during the cross-evaluation as LeNet.

ResNet

The ResNet-like baseline model was implemented as explained in the methodology Section 4.5.4. The final RMSE of the best model was 1.009 for the left hand and 1.012 for the right hand. The model is followed referred during the cross-evaluation as ResNet.

MNet

The MNet model was implemented as explained in the methodology Section 4.6.1. The final RMSE of the best model was 0.832 for the left hand and 0.838 for the right hand. The model was tested with and without the relative power spectrum integration. The added information yield a relevant improvement in the final performance. The final RMSE was 0.424 for the left hand and 0.460 for the right hand. The two different models are referred later in the thesis as follow:

- MNet for normal implementation.
- RPS-MNet for the implementation with the relative power spectrum integration.

SCNN

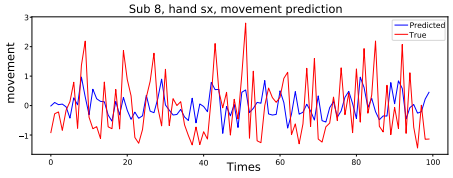
The SCNN model was implemented as explained in the methodology Section 4.6.2. The final RMSE of the best model was 0.947 for the left hand and 0.953 for the right hand. The model was tested with and without the relative power spectrum integration. RPS integration yielded better performance than the basic version. The best model with RPS integration got an RMSE of 0.873 for the left hand and 0.842 for the right hand. The model is following referred during the cross-evaluation as follow:

- SCNN for normal implementation.
- RPS-SCNN for the implementation with the relative power spectrum integration.

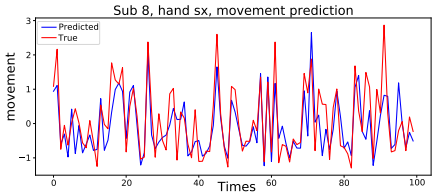
	SPoC	RPS MLP	LeNet	ResNet	SCNN	RPS SCNN	MNet	RPS MNet
Left hand	0.976	0.779	0.975	1.009	0.947	0.873	0.832	**0.428**
Right hand	0.965	0.782	0.978	1.012	0.953	0.841	0.838	**0.460**

Table 12: Summary of Subject 8 both hand results RMSE comparison.

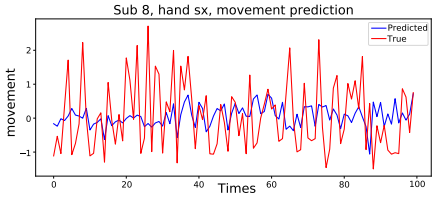
Table 12 shows the comparison between all the models on both the hands. There was no substantial difference between the two hands such that each model can equally predict both the hands. Moreover, it shows that the best solution was MNet with RPS integration since it was significantly outperforming all the other models on the Subject 8 experiments performed. Figure 20 shows a visual evaluation of all the models predicting the target variable.



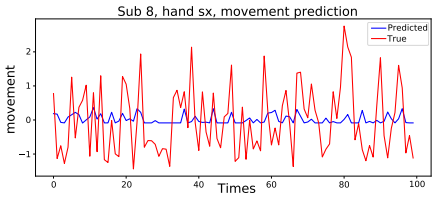
(a) SPoC (fix dimension)



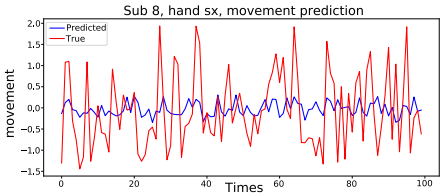
(b) RPS-MLP



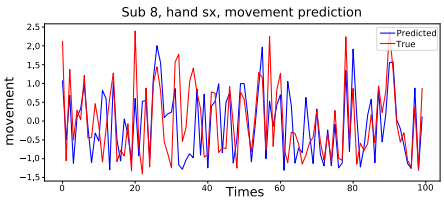
(c) ResNet



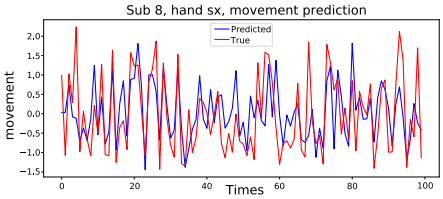
(d) LeNet



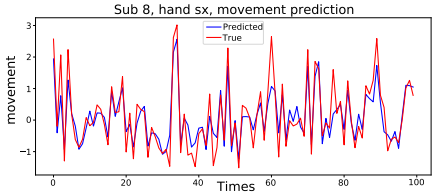
(e) SCNN



(f) RPS-SCNN



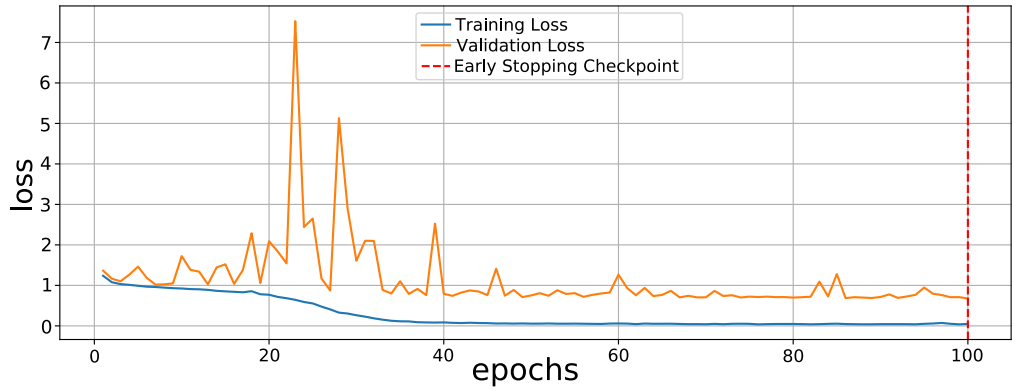
(g) MNet



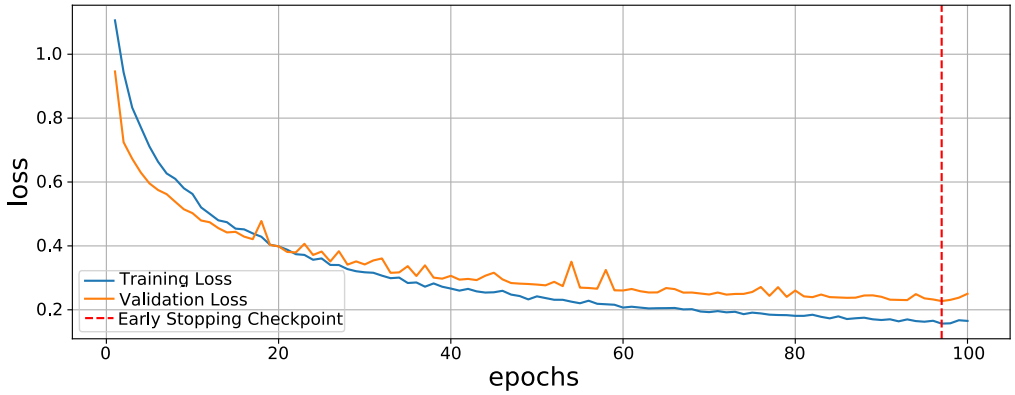
(h) RPS-MNet

Figure 20: Y expected (Red) against Y predicted (Blue). The plots consist of comparing the predicted values and the actual true values of the target hand-kinematic parameter. Specifically, the plots contain 100 predicted y (Blue) values against the y true values (Red). As a result, being the prediction rate of 200 ms, the plots show a prediction of 20 seconds of the hand targeted parameter. The plotted test set was narrowed down to enhance the visualization

5.1.2 RPS integration analysis



(a) MNet



(b) RPS-MNet

Figure 21: MNet versus RPS-MNet loss analysis. This figure analyze the difference between MNet with and without relative power spectrum integration. They show the training and validation losses along the epochs. The dashed red line is the final best model of the specific training session.

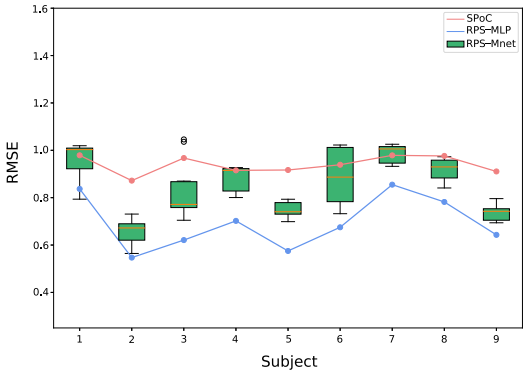
5.1.3 Subjects comparison

Figures 22 and 23 show that the trend displayed in Subject 8 is replicated in all the others while comparing the models between each other. For instance, the RPS-MNet substantially outperformed the other models in all the subjects.

RPS-SCNN

Sub.	SPoC	RPS MLP	RPS SCNN
1	0.979	0.837	0.794
2	0.872	0.547	0.565
3	0.967	0.621	0.705
4	0.915	0.702	0.801
5	0.917	0.575	0.699
6	0.939	0.675	0.732
7	0.979	0.855	0.932
8	0.976	0.782	0.841
9	0.911	0.643	0.694

(a)



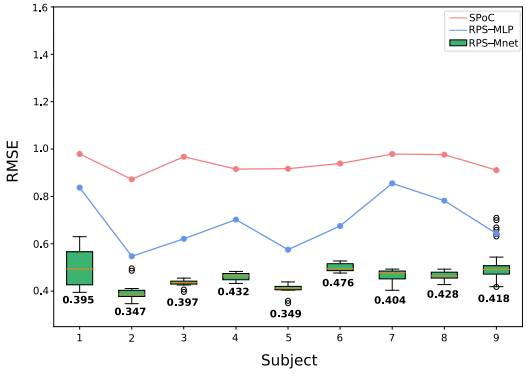
(b)

Figure 22: Multi-subject analysis of RPS-SCNN compared to SPoC and RPS-MLP. Table a) summarize the best model RMSE of the three analyzed methods for all the subjects. Plot b) visually display the three analyzed methods RMSEs. Boxplots represent the RPS-SCNN to show the variance between all the trained models during parameter tuning process. The light red line represents the SPoC algorithms, and, finally, the light blue line represents the RPS-MLP.

RPS-MNet

Subject	SPoC	RPS MLP	RPS MNet
1	0.979	0.837	0.395
2	0.872	0.547	0.347
3	0.967	0.621	0.397
4	0.915	0.702	0.432
5	0.917	0.575	0.349
6	0.939	0.675	0.476
7	0.979	0.855	0.404
8	0.976	0.782	0.428
9	0.911	0.643	0.418

(a)



(b)

Figure 23: Multi-subject analysis of RPS-MNet compared to SPoC and RPS-MLP. Table a) summarize the best model RMSE of the three analyzed methods for all the subjects. The plot b) visually display the three analyzed methods RMSEs. Boxplots represent the RPS-MNet to show the variance between all the trained models during parameter tuning. The light red line represents the SPoC algorithms, and, finally, the light blue line represents the RPS-MLP.

5.2 ECoG experiments

This section will introduce the auxiliary results generated on the ECoG dataset. They have been generated to test the proposed solutions with a smaller dataset to perform a similar task. The detailed comparison between the two datasets is explained in Section 3.3. In these validation experiments, only the results on one subject are reported.

5.2.1 Focus on a sample Subject

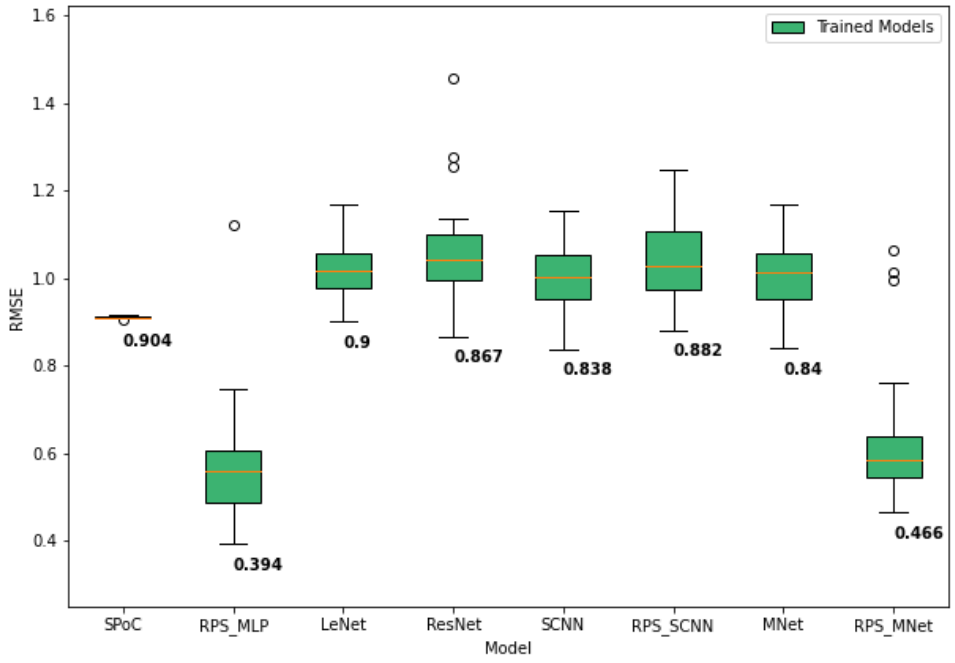
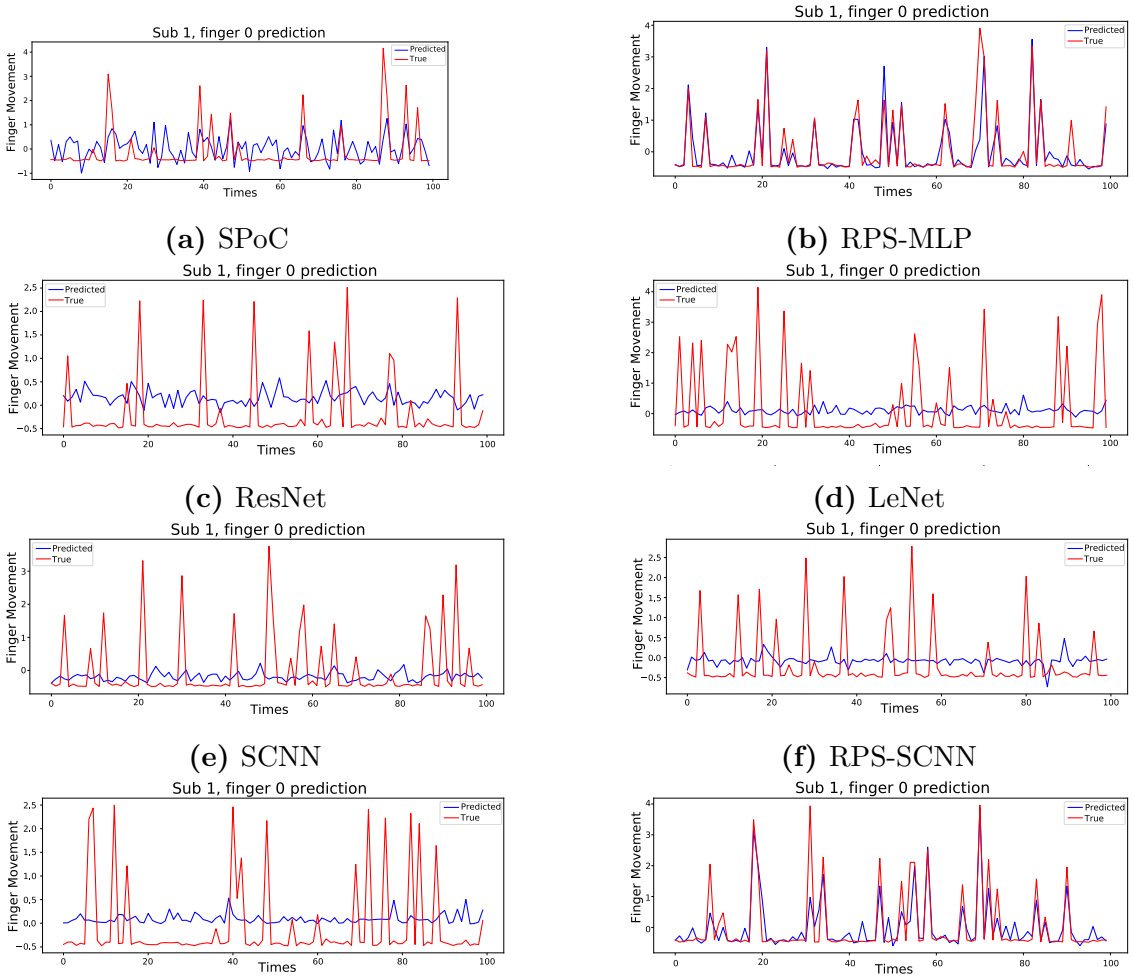


Figure 24: Models comparison. The plot represent the RMSE model comparison on Subject 1, predicting the finger flexion of the thumb. The box-plots show the variance between the different models trained during the hyper-parameter tuning process. The number on the bottom of each box-plot is the RMSE of the best model.



(a) SPoC

(b) RPS-MLP

(c) ResNet

(d) LeNet

(e) SCNN

(f) RPS-SCNN

(g) MNet

(h) RPS-MNet

Figure 25: Y expected (Red) against Y predicted (Blue). The plots consist of comparing the predicted values and the actual true values of the target finger flexion. Specifically, the plots contain 100 predicted y (Blue) values against the y true values (Red). As a result, being the prediction rate of 200 ms, the plots show a prediction of 20 seconds of the targeted parameter. The plotted test set was narrowed down to enhance the visualization

6 Discussion

This section will discuss the experiment results and findings from chapter 5. The focus of this thesis was to explore the potentiality of deep Learning decoding hand-kinematics parameters. Specifically, it aimed to replay the following questions:

- Can deep learning decode continuous hand-kinematics parameters from within-subject MEG measurements?
- How precisely can deep learning decode continuous hand-kinematics parameters compared to state-of-the-art techniques?

MEG measurements are complex data structures with a low signal-to-noise ratio. Additionally, the hand-kinematics parameter analyzed was a continuous variable. As a result, the problem was a complex regression task. To optimally tackle the problem, CNN-based models were chosen as DL solutions due to their ability to extract complex patterns from raw input data. Table 12 shows no relevant difference in predicting the two hand parameters such that all the models got similar performance in both hands. Therefore, the following conclusion can be generalized for each hand.

The first point of focus was to understand if DL models were able to perform the decoding task. This analysis has been evaluated in two different steps: first of all, two well-known architectures were chosen as baseline models to understand the potentiality of general CNN designed not specifically to work with MEG. Second, two CNN-based architectures designed to work with MEG recording were tested. The results show that the general CNN architectures could not perform the task accurately as shown in the respective Figures 20c and 20d. On the other hand, Figure 20h shows that RPS-MNet performed the task optimally. As a result, deep learning can be considered an option in the sector as far as the models are specifically designed to extract features from the MEG recordings.

To fully understand the potential of deep learning, the second point of focus of the analysis was to compare the DL solutions with the supervised spatial filtering algorithm Source Power Comodulation, a technique that stands out in performing regression tasks. The results (Table 12) show that all the proposed solution outperformed the SPoC algorithms. However, only RPS-MNet substantially got an improvement in the performances. The final RMSEs of the best models on Subject 8 were 0.424 for the left hand and 0.460 for the right hand compared to the initial RMSEs of the SPoC algorithm of 0.976 for the left hand and 0.965 for the right hand. Additionally, Figures 20a and 20h graphically show that the proposed model can approximate the target parameter considerably more accurately in comparison with SPoC. Overall, regarding the baseline models, the SPoC algorithms could not reach good predictions on the analyzed task. On the other hand, the RPS-MLP baseline

model proposed presented a valid solution compared to the other models since it was the second-best performing model on both hands.

Furthermore, the two proposed architectures (MNet and SCNN) are designed to extract specific relevant features from the MEG input, such that they tend to mimic sector-based feature extraction pipelines. MNet is designed to extract global features from the input data convolving on both channel and time domains. Next, it extracts features along the time domain to derive frequency-based features, and, eventually, it performs convolutions with small kernels to extract local features. On the other hand, SCNN is designed to extract features from the MEG measurement in a specific way such that first, the data are spatially filtered, therefore the feature extraction is applied across the channel domain only. Second, it performs temporal filtering such that filters are applied only on the time domain. Thus, the convolution is applied in the channel and temporal domains separately. Since MNet got better results in all the experiments performed, it is fairly to conclude that the network is more inclined to learn better from spatio-temporal features analyzed simultaneously.

Regarding the auxiliary experiments, Figure 24 shows that the best model is the baseline RPS-MLP. The second-best model is the proposed solution RPS-MNet. Figure 25 also proves that the SPoC algorithms and the general CNN-based architectures do not optimally perform the task. In addition, it proves that, even though the basic version of the proposed models (SCNN and MNet) got better performances than the SPoC algorithm, they did not perform the task optimally. As a result, the Relative Power Spectrum can extract meaningful features from a smaller dataset. On the other hand, this additional experiment specifically shows that DL solutions are data hungrier.

Overall, Figure 20h shows that the proposed solution RPS-MNet can approximate the target value with good performances also in the ECoG experiments.

6.1 Multi-subject analysis

The proposed solution integrated with the RPS and the principal baseline models were additionally evaluated on eight different subjects. The detected trends on Subject 8 were replicated across all the subjects, i.e., the RPS-MNet considerably outperformed all the other models in all the subjects. However, Figures 22b and 23b show that if focusing on each model separately, the results of each model vary from subject to subject and the trend is generally replicated throughout all the models. Indeed, for instance, all the models got the smallest RMSE value in Subject 2. From this interpretation is reasonable to conclude that the goodness of the prediction is affected by intra-subject variability of the MEG recording. This subject variability is usually generated by differences in neural mechanisms and brain anatomy (Ou et al. 2007).

6.2 Relative Power Spectrum integration

The relative power spectrum integration boosted the performances of the deep learning models as well as, while combined with a simple MLP, it reached the second-best performances on predicting the target in the principal experiments. Regarding the SCNN, on Subject 8, the integration reduced the RMSE from 0.947 to 0.873 for the left hand and from 0.947 to 0.841 for the right hand. Moreover, the MNet performances increased in the version with the integration from an RMSE of 0.832 to 0.428 for the left hand and from 0.838 to 0.460 for the right hand. Consequently, it is reasonable to conclude that the relative power spectrum can extract relevant features from the MEG recording that are correlated to the hand-kinematic target parameter.

In MNet, the RPS integration helped to regularize the training of the model. On the one hand, in Figure 21a, the training error decreases while the validation loss does not decrease properly. A trend that can be caused by overfitting the training dataset. On the other hand, in Figure 21b both the training and validation losses are constantly decreasing; thus, the model could eventually generalize optimally on the test set. Two possible interpretations can be highlighted from the RPS integration analysis.

First of all, the dataset may be too small such that the DL solutions may require more data to fully exploit their potentiality. Indeed, the additional information provided by the RPS integration helped the models to generalize better. To additionally support this conclusion, all the DL-based solutions tested more or less overfitted the trained dataset, even though regularization techniques such as dropout, data augmentation, and weight normalization were applied. Second, given that alpha waves mainly encode the hand-kinematic, the RPS integration added relevant information to the model such that the network can use the different band power values to understand, for example, when the movement is happening or not. As a result, the model can tune the weights accordingly. Besides, the RPS may add precious information regarding artifacts or environmental noise as well.

Regarding the auxiliary experiments, on the one hand, the RPS integration did not boost up the performances on the RPS-SCNN solution. On the other hand, MNet was boosted up from an RMSE of 0.84 to an RMSE of 0.466 got by the RPS-MNet version. However, the RPS-MNet got worst performances than the simpler RPS-MLP. This result may be explained by the fact that the dataset size during this experiment was very small. Indeed, the additional complexity of the proposed models had a negative effect on the generalization properties. Besides, it is fair to conclude that the RPS works properly independently for the dataset size.

Finally, the last conclusion can be extracted by analyzing the difference in performances between the RPS-MLP during the two experiments. Even though the

target parameter is different, in the ECoG experiment, Figure 25b shows that the model can predict the target optimally. On the other hand, Figure 20b shows that the model was not performing as good as in the ECoG one. This result may be explained by the fact that being the SNR higher in ECoG compared to MEG, the RPS can extract more meaningful features from the recording.

6.3 Dataset size

Due to the novelty and particularity of the task, it is impossible to have a straightforward comparison of the dataset size used in the main experiments with other similar studies. However, some analysis and some consideration can be carried out. Based on the studies of Kostas and colleagues 2019 and Aoe and colleagues 2019, whereby the two proposed solution were inspired, the primary datasets used during the experiments consisted of approximately 700–1000 minutes of recordings divided into multiple subjects. To point out that the tasks performed were between-subject; therefore, all the available recordings could have been used. Consequently, even though the task was different, the MEG dataset size available to train the network in both the reference studies was one order of magnitude larger than the one used in the within-subject experiments performed in the thesis. Moreover, the analysis of the loss function during the training of the proposed solutions showed a general overfitting trend to the training data. Besides, the fact that the DL solutions were not able to improve the performances of the simple RPS-MLP solution in the auxiliary experiment Section (5.2), in which the dataset was even smaller. It is reasonable to conclude that the proposed DL solutions may benefit from an increase in the size of the dataset. Therefore, they may yield even better performances with more training data. Specifically, the SCNN may benefit the most from the size increasing since it is a more complex architecture than MNet.

On the other hand, recording new MEG data is complex, specifically with unhealthy subjects. As a result, finding a solution that can learn the analyzed task with a limited number of data is definitely advantageous. Therefore, the good performances achieved by the RPS-MNet proposed solution with the limited training data gain more relevance.

To conclude, in general, the complexity of the recording of new data and the variability and complexity of the MEG recording make the concept of open data a critical factor in the rise of deep learning applications in the sector.

6.4 Future work

The DL solutions proposed outperformed the state-of-the-art model tested. However, the main experiment conducted represents an initial evaluation. Therefore, to fully understand the potentiality of deep learning to become a reliable concrete alternative in real decoding applications such as hand-movement-based BCI, additional studies should be performed. The following future works are proposed to evaluate the proposed solutions further.

First, try modeling with more training data. This evaluation can be performed moving to a between-subject experiment such that the model can be trained with all the different subjects recording and, eventually, test it on a held-out subject. However, more variability inside the recording will be introduced to the task, such as inter-subject and inter-visit variability. As a result, the robustness of the solutions will be additionally tested.

Second, all the subjects that performed the experiment were healthy. Indeed, the actual movement was performed. However, it would be interesting to test the solutions to a similar imagery task such that the solution can be used to help unhealthy subjects. In this way, the model may be more relevant for future BCI applications, for example, helping during the rehabilitation of some patients.

Third, a combination with the CNN based porpoised network and RNN can further be tested. This combined solution is currently already implemented in multiple sector studies (Roy et al. 2019). The RNN integration may boost the performances since it can analyze long-term time features from the input signals.

Finally, thinking about potential new recordings generated by new experiments to augment the data can be interesting to use different neuroimaging sources. As a result, the MEG recording is combined with other neuroimaging techniques to create a multi-modal data input. However, this cannot be done with the currently available primary dataset since it is composed of only MEG recording.

References

Abadi, Martín et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems". In: *arXiv:1603.04467 [cs]* (Mar. 2016). arXiv: 1603.04467. URL: http://arxiv.org/abs/1603.04467.

Aoe, Jo et al. "Automatic diagnosis of neurological diseases using MEG signals with a deep neural network". en. In: *Scientific Reports* 9.1 (Mar. 2019). Number: 1 Publisher: Nature Publishing Group, p. 5057. ISSN: 2045-2322. DOI: 10.1038/s41598-019-41500-x.

Baillet, Sylvain. "Magnetoencephalography for brain electrophysiology and imaging". en. In: *Nature Neuroscience* 20.3 (Mar. 2017). Number: 3 Publisher: Nature Publishing Group, pp. 327–339. ISSN: 1546-1726. DOI: 10.1038/nn.4504.

Batula, Alyssa M. et al. *Comparison of Brain Activation during Motor Imagery and Motor Movement Using fNIRS*. en. Research Article. ISSN: 1687-5265 Library Catalog: www.hindawi.com Pages: e5491296 Publisher: Hindawi Volume: 2017. May 2017. DOI: https://doi.org/10.1155/2017/5491296.

Benedetti, Fabrizio. "Sensory and Motor Functions of the Hand". en. In: *Writing Systems and Cognition: Perspectives from Psychology, Physiology, Linguistics, and Semiotics*. Ed. by W. C. Watt. Neuropsychology and Cognition. Dordrecht: Springer Netherlands, 1994, pp. 347–373. ISBN: 978-94-015-8285-8. DOI: 10.1007/978-94-015-8285-8_17.

Bergstra, James and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13.10 (2012), pp. 281–305. ISSN: 1533-7928. URL: http://jmlr.org/papers/v13/bergstra12a.html.

Bisley, James W. "Parietal Lobe". en. In: *Encyclopedia of Animal Cognition and Behavior*. Ed. by Jennifer Vonk and Todd Shackelford. Cham: Springer International Publishing, 2017, pp. 1–5. ISBN: 978-3-319-47829-6. DOI: 10.1007/978-3-319-47829-6_1252-1.

Branco, Mariana P. et al. "Encoding of kinetic and kinematic movement parameters in the sensorimotor cortex: A Brain-Computer Interface perspective". en. In: *European Journal of Neuroscience* 50.5 (2019), pp. 2755–2772. ISSN: 1460-9568. DOI: 10.1111/ejn.14342.

Bunge, S.A. and I. Kahn. "Cognition: An Overview of Neuroimaging Techniques". en. In: *Encyclopedia of Neuroscience*. Elsevier, 2009, pp. 1063–1067. ISBN: 978-0-08-045046-9. DOI: 10.1016/B978-008045046-9.00298-9.

Chen, Andrew et al. "Developments in MLflow: A System to Accelerate the Machine Learning Lifecycle". In: *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning*. DEEM'20. Portland, OR, USA: Association for Computing Machinery, June 2020, pp. 1–4. ISBN: 978-1-4503-8023-2. DOI: 10.1145/3399579.3399867.

Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter. "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". In: *arXiv:1511.07289 [cs]* (Feb. 2016). URL: http://arxiv.org/abs/1511.07289.

CNX OpenStax. *DIfferent regions of the motor cortex control different muscle groups and their neighbouring regions control the neighbouring muscle groups.* 2016. URL: http://cnx.org/contents/GFy_h8cu@10.53:rZudN6XP@2/Introduction.

Cvlr, Robin. *CNN general architecture.* 2017. URL: https://github.com/robincvlr/grpa-usher/wiki/4.-Strat%C3%A9gie-de-classification.

Dähne, Sven et al. "SPoC: A novel framework for relating the amplitude of neuronal oscillations to behaviorally relevant parameters". en. In: *NeuroImage* 86 (Feb. 2014), pp. 111–122. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2013.07.079.

David Baillot. *Illustration of a neuron.* 2018. URL: https://medicalxpress.com/news/2018-07-neuron-axons-spindly-theyre-optimizing.html.

Ebner, Timothy J., Claudia M. Hendrix, and Siavash Pasalar. "Past, Present, and Emerging Principles in the Neural Encoding of Movement". In: *Advances in experimental medicine and biology* 629 (2009), pp. 127–137. ISSN: 0065-2598. DOI: 10.1007/978-0-387-77064-2_7.

Glosser.ca. *Artificial neural network with layer coloring.* 2013.

Gramfort, Alexandre et al. "MEG and EEG data analysis with MNE-Python". English. In: *Frontiers in Neuroscience* 7 (2013). Publisher: Frontiers. ISSN: 1662-453X. DOI: 10.3389/fnins.2013.00267.

Gramfort, Alexandre et al. "MNE software for processing MEG and EEG data". en. In: *NeuroImage* 86 (Feb. 2014), pp. 446–460. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2013.10.027.

Hämäläinen, Matti et al. "Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain". en. In: *Reviews of Modern Physics* 65.2 (Apr. 1993), pp. 413–497. ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.65.413.

He, Kaiming et al. "Deep Residual Learning for Image Recognition". In: (Dec. 2015). URL: http://arxiv.org/abs/1512.03385.

Herculano-Houzel, Suzana. "The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost". en. In: *Proceedings of the National Academy of Sciences* 109.Supplement 1 (June 2012). Publisher: National Academy of Sciences Section: Colloquium Paper, pp. 10661–10668. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1201895109.

Hoerl, Arthur E. and Robert W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems". In: *Technometrics* 12.1 (Feb. 1970), pp. 55–67. ISSN: 0040-1706. DOI: 10.1080/00401706.1970.10488634.

Ioffe, Sergey and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *arXiv:1502.03167 [cs]* (Mar. 2015). arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

Jerbi, Karim et al. "Coherent neural representation of hand speed in humans revealed by MEG imaging". en. In: *Proceedings of the National Academy of Sciences* 104.18 (May 2007). Publisher: National Academy of Sciences Section: Biological Sciences, pp. 7676–7681. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0609632104.

Jillian Higgins. *Brain Illustration*. 2017. URL: https://braintumor.org/brain-tumor-information/signs-and-symptoms/brain-illustration/.

Khan, Salman et al. "A Guide to Convolutional Neural Networks for Computer Vision". In: *Synthesis Lectures on Computer Vision* 8.1 (Feb. 2018). Publisher: Morgan & Claypool Publishers, pp. 1–207. ISSN: 2153-1056. DOI: 10.2200/S00822ED1V01Y201712COV015.

Kiefer, J. and J. Wolfowitz. "Stochastic Estimation of the Maximum of a Regression Function". EN. In: *Annals of Mathematical Statistics* 23.3 (Sept. 1952). Publisher: Institute of Mathematical Statistics, pp. 462–466. ISSN: 0003-4851, 2168-8990. DOI: 10.1214/aoms/1177729392.

Kingma, Diederik P. and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv:1412.6980 [cs]* (Jan. 2017). arXiv: 1412.6980. URL: http://arxiv.org/abs/1412.6980.

Klambauer, Günter et al. "Self-Normalizing Neural Networks". In: *arXiv:1706.02515 [cs, stat]* (Sept. 2017). arXiv: 1706.02515. URL: http://arxiv.org/abs/1706.02515.

Kolasinski, James et al. "Spatially and temporally distinct encoding of muscle and kinematic information in rostral and caudal primary motor cortex". en. In: *bioRxiv* (Oct. 2019). Publisher: Cold Spring Harbor Laboratory Section: New Results, p. 613323. DOI: 10.1101/613323.

Kostas, Demetres, Elizabeth W. Pang, and Frank Rudzicz. "Machine learning for MEG during speech tasks". en. In: *Scientific Reports* 9.1 (Feb. 2019). Number: 1 Publisher: Nature Publishing Group, p. 1609. ISSN: 2045-2322. DOI: 10.1038/s41598-019-38612-9.

LeCun, Y. et al. "Backpropagation Applied to Handwritten Zip Code Recognition". en. In: *Neural Computation* 1.4 (Dec. 1989), pp. 541–551. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/neco.1989.1.4.541.

Lisi, G. and J. Morimoto. "Chapter Seven - Noninvasive Brain Machine Interfaces for Assistive and Rehabilitation Robotics: A Review". en. In: *Human Modelling for Bio-Inspired Robotics*. Ed. by Jun Ueda and Yuichi Kurita. Academic Press, Jan. 2017, pp. 187–216. ISBN: 978-0-12-803137-7. DOI: 10.1016/B978-0-12-803137-7.00006-9.

Liu, Danqing. *A Practical Guide to ReLU*. 2017. URL: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7.

Mike X Cohen. *Neuroscience as source separation*. 2019. URL: https://www.youtube.com/watch?v=ukjuFUghieg&list=PLn0OLiymPak0t1moK3sn4Sl1seXlEOPHT.

Miller, Kai J and Gerwin Schalk. "Prediction of Finger Flexion 4th Brain-Computer Interface Data Competition". en. In: (2008), p. 2.

Nair, Vinod and Geoffrey E Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". en. In: (2010), p. 8.

Ou, Wanmei, Polina Golland, and Matti Hamalainen. "Sources of Variability in MEG". en. In: (2007), p. 8.

Paszke, Adam et al. "Automatic differentiation in PyTorch". en. In: (Oct. 2017). URL: https://openreview.net/forum?id=BJJsrmfCZ.

Pedregosa, Fabian et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. URL: http://jmlr.org/papers/v12/pedregosa11a.html.

Pfurtscheller, G. and A. Aranibar. “Event-related cortical desynchronization detected by power measurements of scalp EEG”. eng. In: *Electroencephalography and Clinical Neurophysiology* 42.6 (June 1977), pp. 817–826. ISSN: 0013-4694. DOI: 10.1016/0013-4694(77)90235-8.

Pfurtscheller, G. and C. Neuper. “Motor imagery activates primary sensorimotor area in humans”. eng. In: *Neuroscience Letters* 239.2-3 (Dec. 1997), pp. 65–68. ISSN: 0304-3940. DOI: 10.1016/s0304-3940(97)00889-6.

Pfurtscheller, G., A. Stancák, and Ch. Neuper. “Event-related synchronization (ERS) in the alpha band — an electrophysiological correlate of cortical idling: A review”. en. In: *International Journal of Psychophysiology*. New Advances in EEG and cognition 24.1 (Nov. 1996), pp. 39–46. ISSN: 0167-8760. DOI: 10.1016/S0167-8760(96)00066-9.

Rickert, Jörn et al. “Dynamic Encoding of Movement Direction in Motor Cortical Neurons”. en. In: *Journal of Neuroscience* 29.44 (Nov. 2009). Publisher: Society for Neuroscience Section: Articles, pp. 13870–13882. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.5441-08.2009.

Rosenblatt, F. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* (1958). DOI: 10.1037/H0042519.

Roy, Yannick et al. “Deep learning-based electroencephalography analysis: a systematic review”. eng. In: *Journal of Neural Engineering* 16.5 (2019), p. 051001. ISSN: 1741-2552. DOI: 10.1088/1741-2552/ab260c.

Sabbagh, David et al. “Manifold-regression to predict from MEG/EEG brain signals without source modeling”. In: *arXiv:1906.02687 [cs, eess, stat]* (Nov. 2019). arXiv: 1906.02687. URL: http://arxiv.org/abs/1906.02687.

Sabbagh, David et al. “Predictive regression modeling with MEG/EEG: from source power to signals and cognitive states”. en. In: *NeuroImage* (May 2020), p. 116893. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2020.116893.

Santello, Marco, Gabriel Baud-Bovy, and Henrik Jörntell. “Neural bases of hand synergies”. In: *Frontiers in Computational Neuroscience* 7 (Apr. 2013). ISSN: 1662-5188. DOI: 10.3389/fncom.2013.00023.

Schalk, G. et al. “Decoding two-dimensional movement trajectories using electrocorticographic signals in humans”. eng. In: *Journal of Neural Engineering* 4.3 (Sept. 2007), pp. 264–275. ISSN: 1741-2560. DOI: 10.1088/1741-2560/4/3/012.

Singh, Sanjay P. “Magnetoencephalography: Basic principles”. en. In: *Annals of Indian Academy of Neurology* 17.5 (Mar. 2014). Company: Medknow Publications and Media Pvt. Ltd. Distributor: Medknow Publications and Media Pvt. Ltd. Institution: Medknow Publications and Media Pvt. Ltd. Label: Medknow Publications and Media Pvt. Ltd. Publisher: Medknow Publications, p. 107. ISSN: 0972-2327. DOI: 10.4103/0972-2327.128676.

Srivastava, Nitish et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–

1958. ISSN: 1533-7928. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Tangermann, Michael et al. "Review of the BCI Competition IV". English. In: *Frontiers in Neuroscience* 6 (2012). Publisher: Frontiers. ISSN: 1662-453X. DOI: 10.3389/fnins.2012.00055.

Taulu, Samu, Matti Kajola, and Juha Simola. "Suppression of Interference and Artifacts by the Signal Space Separation Method". en. In: *Brain Topography* 16.4 (June 2004), pp. 269–275. ISSN: 1573-6792. DOI: 10.1023/B:BRAT.0000032864.93890.f9.

Tokozume, Yuji and Tatsuya Harada. "Learning environmental sounds with end-to-end convolutional neural network". en. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, LA: IEEE, Mar. 2017, pp. 2721–2725. ISBN: 978-1-5090-4117-6. DOI: 10.1109/ICASSP.2017.7952651.

Van Rossum, Guido and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 978-1-4414-1269-0.

Welch, P. "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms". In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (June 1967). Conference Name: IEEE Transactions on Audio and Electroacoustics, pp. 70–73. ISSN: 1558-2582. DOI: 10.1109/TAU.1967.1161901.

Zhang, Yu-Dong et al. "Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation". en. In: *Information Fusion* 64 (Dec. 2020), pp. 149–187. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2020.07.006.